TPC BENCHMARK™ W

(Web Commerce)

Draft Specification

Revision D-5.1

August 18, 1999

Transaction Processing Performance Council (TPC)
777 N. First Street, Suite 600
San Jose, CA 95112-6311, USA
Phone (408) 295-8894
Fax (408) 295-9768
http://www.tpc.org
e-mail:admin@tpc.org

Acknowledgments

The TPC acknowledges the substantial contribution of …

TPC Membership

(as of January, 1999)

Amdahl
ALR
BEA
Bull S.A.
Compaq Computer Corp
Data General Corp.
Dell
EDS
EMC Corp.
Fujitsu / ICL
Hewlett-Packard
Hitachi SW
IBM Corp.
IDEAS International
Information Decisions
Informix Software
Intel Corp.
Intergraph
ITOM International
Microsoft Corp.
Mitsubishi Electric Corp.
Motorola
NCR
NEC Systems Laboratory
Nikkei Business Publications
OKI Electric Industry
Olivetti S.p.A
Oracle Corp.
Samsung
SCO
Sequent Computer
Siemens Nixdorf Information
Silicon Graphics
Software AG
Stratus Computer
Sun Microsystems
Sybase
Toshiba Corp.
Tricord Systems Inc.
Unisys Corp.
(incomplete list)

Trademarks and Legal Notices

TPC Benchmark™, TPC-W, WIPS, WIPSo and WIPSb are trademarks of the Transaction Processing Performance Council.

Document History

Date	Version	Description
June 6, 1997	Draft 0.1	Workgroup Draft
August 27, 1997	Draft 0.2	Schema, Metrics, RTE, Transactions updated
Sept. 10, 1997	Draft 0.3	Fully cleaned up
October 16, 1997	Draft 0.4	Additions to Database Population, Schema, transaction.
Nov. 30, 1997	Draft 0.5	Addition of Order Status
December 2, 1997	Draft 0.6	Rework the Scaling, clean out ACID and Performance Metrics.
December 3, 1997	Draft 0.7	More modifications
June 18, 1998	Draft 1.0	Schema changes
July 14, 1998	Draft 1.1	Clause 0 - SET
August 4, 1998	Draft 1.2	Cleanup
Sept. 8, 1998	Draft 1.3	Clause 1,2 and Clause 4 changes
Sept. 10, 1998	Draft 1.4	Clause 2 updates
Sept. 29, 1998	Draft 1.5	Clause 3 and 4 changes
October 12, 1998	Draft 1.6	Clause 1 and 4 changes
October 13, 1998	Draft 1.7	Clause 6 changes
October 15, 1998	Draft 1.8	First complete draft
January 12, 1999	Draft 2.0	Major draft revision
March 5, 1999	Draft 3.0	Major draft revision
March 23, 1999	Draft 3.1	Draft revision following Orange County Meeting
April 2, 1999	Draft 3.2	Draft revision prior to Colorado Springs meeting
April 22, 1999	Draft 3.3	Draft revision following Colorado Springs meeting
May 30, 1999	Draft 4.0	Draft revision following Denver meeting
June 16, 1999	Draft 4.1	Draft revision prior to Grenoble meeting
July 5, 1999	Draft 4.2	Draft revision following Grenoble meeting
July 12, 1999	Draft 5.0	Public and Member Company Review Draft

Table of Contents

Clause 0 - PREAMBLE

0.1 Introduction

TPC Benchmark™ W (TPC-W) is a transactional web benchmark. The workload is performed in a controlled Internet Commerce environment that simulates the activities of a business oriented Web Server. The workload exercises a breadth of system components associated with such environments, which are characterized by:

- Multiple on-line browser sessions
- Dynamic page generation with database access and update
- Consistent web objects
- The simultaneous execution of multiple transaction types that span a breadth of complexity
- On-line transaction execution modes
- Databases consisting of many tables with a wide variety of sizes, attributes, and relationships
- Transaction integrity (ACID properties)
- Contention on data access and update

The performance metric reported by TPC-W is a "business throughput" measuring the number of web interactions processed per second. Multiple transactions are used to simulate the business activity of processing an order, and each transaction is subject to a response time constraint. The performance metric for this benchmark is expressed in Web Interactions per second (WIPS).

TPC-W simulates three different profiles by varying the ratio of browse to buy: primarily shopping (WIPS), browsing (WIPSb) and web-based ordering (WIPSo). All references to WIPS (WIPSb, WIPSo) results must include the primary metrics which are: the WIPS rate, the associated price per WIPS ($/WIPS), and the availability date of the priced configuration.

The following functions must be implemented using commercially available products.

- Operating System
- Web Server (HTTP Server)
- Database Server

To implement the electronic commerce function one may use commercially available products or implementation specific programs.

The electronic commerce function must include, at minimum, the following capabilities as defined in this specification:

- Secure Socket Layer (SSL)
- Shopping Cart
- Credit Card Verification
- Secure on-line payment authorization

Although these specifications express implementation in terms of a relational data model with a conventional locking scheme, the database may be implemented using any commercially available database management system (DBMS), database server, file system, or other data repository that provides a functionally equivalent implementation. The terms "table", "row", and "column" are used in this document only as examples of logical data structures.

TPC-W uses terminology and metrics that are similar to other benchmarks originated by the TPC or others. Such similarity in terminology does not in any way imply that TPC-W results are comparable to other benchmarks. The only benchmark results comparable to TPC-W is another TPC-W result with the same revision.

Despite the fact that this benchmark offers a rich environment that emulates many web browsing and web-based ordering applications, this benchmark does not reflect the entire range of Web Server requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-W approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to any other environment are not recommended.

Benchmark results are highly dependent upon workload, specific application requirements, systems design, and implementation. Relative system performance will vary as a result of these and other factors. Therefore, TPC-W should not be used as a substitute for a specific customer application benchmarking when critical capacity planning and / or product evaluation decisions are contemplated.

Benchmark sponsors are permitted several possible system designs, insofar as they adhere to the model described and pictorially illustrated in Clause 6. A Full Disclosure Report of the implementation details, as specified in Clause 8, must be made available along with the reported results.

Comment: While separated from the main text for readability, comments are a part of the standard and are enforced. However, the sample programs included as Appendix A, the summary statements included as Appendix B, and the numerical quantities summary, included as Appendix C, are provided only as examples and are specifically not part of this standard.

0.2 General Implementation Guidelines

The purpose of TPC benchmarks is to provide relevant, objective performance data to industry users. To achieve that purpose, TPC benchmark specifications require that benchmark tests be implemented with systems, products, technologies and pricing that:

- Are generally available to users.
- Are relevant to the market segment that the individual TPC benchmark models or represents (e.g. TPC-W models and represents high-volume, complex web browsing and web-based ordering environments).
- A significant number of users in the market segment the benchmark models or represents would plausibly implement.

The use of new systems, products, technologies (hardware or software), and pricing is encouraged so long as they meet the requirements above. Specifically prohibited are benchmark systems, products, technologies, pricing, and implementations whose primary purpose is performance optimization of TPC benchmark results without any corresponding applicability to real-world applications and environments. In other words, all "benchmark special" implementations that improve benchmark results but not real-world performance or pricing, are prohibited.

The following characteristics should be used as a guide to judge whether a particular implementation is a benchmark special. It is not required that each point below be met, but that the cumulative weight of the evidence be considered to identify an unacceptable implementation. Absolute certainty or certainty beyond a reasonable doubt is not required to make a judgment on this complex issue. The question that must be answered is this: based on the available evidence, does the clear preponderance (the greater share or weight) of evidence indicate that this implementation is a benchmark special.

The following characteristics should be used to judge whether a particular implementation is a benchmark special:

a) Is the implementation generally available, documented, and supported?

b) Does the implementation have significant restrictions on its use or applicability that limits its use beyond TPC benchmarks?

c) Is the implementation or part of the implementation poorly integrated into the larger product?

d) Does the implementation take special advantage of the limited nature of TPC benchmarks (e.g., transaction profile, transaction mix, transaction concurrency and / or contention, transaction isolation) in a manner that would not be generally applicable to the environment the benchmark represents?

e) Is the use of the implementation discouraged by the vendor? (This includes failing to promote the implementation in a manner similar to other products and technologies.)

f) Does the implementation require uncommon sophistication on the part of the end-user, programmer, or system administrator?

g) Is the pricing unusual or non-customary for the vendor or unusual or non-customary to normal business practices? The following pricing practices are suspect:

- Availability of a discount to a small subset of possible customers.
- Discounts documented in an unusual or non-customary manner.
- Discounts that exceed 25% on small quantities and 50% on large quantities.
- Pricing featured as a close-out or one-time special.
- Unusual or non-customary restrictions on transferability of product, warranty or maintenance on discounted items.

h) Is the implementation being used (including beta) or purchased by end-users in the market area the benchmark represents? How many? Multiple sites? If the implementation is not currently being used by end-users, is there any evidence to indicate that it will be used by a significant number of users?

Clause 1 - WEB OBJECT & LOGICAL DATABASE DESIGN

1.1 Business and Application Environment

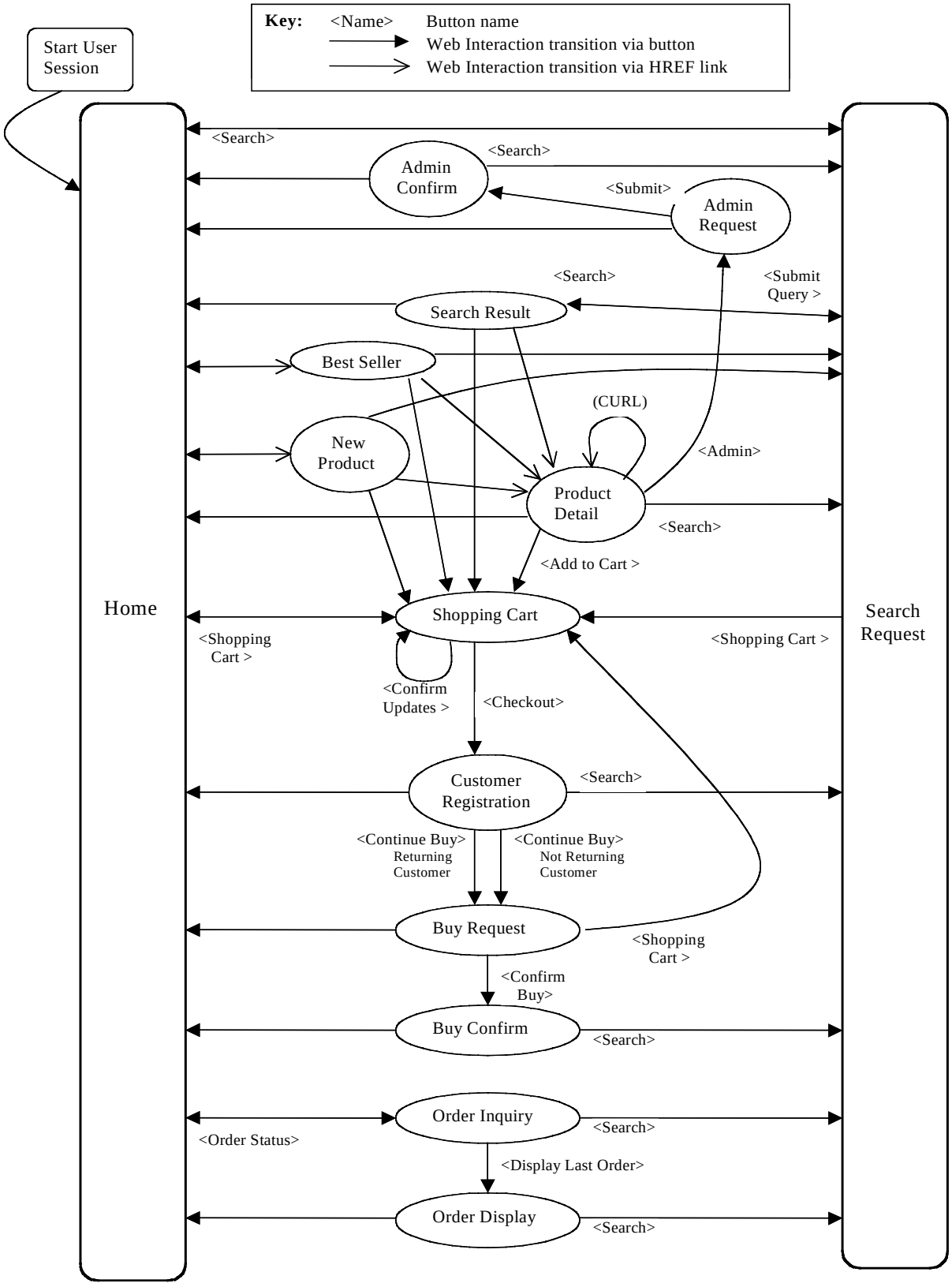
TPC Benchmark™ W comprise a set of basic operations designed to exercise transactional web system functionality in a manner representative of Internet Commerce application environments. These basic operations have been given a real-life context, portraying the activity of a web site that supports user browsing, searching and online ordering activity. This is intended to help users relate intuitively to the components of the benchmark. The workload is centered around the activity of browsing, searching and processing orders and provides a logical database design, which can be distributed without structural changes to the application.

TPC-W does not represent the activity of any particular business segment, but rather any industry which must market and sell a product or service over the Internet (e.g., retail store, software distribution, airline reservation, etc.). TPC-W does not attempt to be a model of how to build an actual application.

The purpose of this benchmark is to reduce the diversity of operations found in an Internet Commerce application, while retaining the application's essential performance characteristics, namely: the level of system utilization and the complexity of operations. A large number of functions have to be performed to manage an environment which supports browse and order processing. We include a representative set of functions. Many other functions are not of primary interest for performance analysis, since they are proportionally small in terms of system resource utilization or in terms of frequency of execution. Although these functions are vital for a production system, they merely create unnecessary diversity in the context of a standard benchmark and have been omitted in TPC-W.

The application portrayed by the benchmark is a Retail Store on the Internet with customer browse and order scenario. Customers visit the Company web site, the store-front, to look at products, find information, place an order, or request the status of an existing order. The majority of visitor activity is to browse the site. Some percent of all visits result in submitting a new order. In addition to using the system as a store-front, it is also used for Administration of the web site. Administration includes modification to the store-front such as price changes, adding and deleting products. The following diagram illustrates the flow of the store front:

Note: The Admin Request and Admin Confirm web interations are administrative tasks. They are not components that a regular customer would see, but exist as part of the transaction model depicted by the workload.


Key: <Name> Button name
Web Interaction transition via button
Web Interaction transition via HREF link
Start User Session
Home
Search Request
<Search>
Admin Confirm
<Search>
<Submit>
Admin Request
<Search>
<Submit Query >
Search Result
Best Seller
(CURL)
New Product
<Admin>
Product Detail
<Search>
<Add to Cart >
Shopping Cart
<Shopping Cart >
<Shopping Cart >
<Confirm Updates >
<Checkout>
Customer Registration
<Search>
<Continue Buy> Returning Customer
<Continue Buy> Not Returning Customer
Buy Request
<Shopping Cart >
<Confirm Buy>
Buy Confirm
<Search>
Order Inquiry
<Search>
<Order Status>
<Display Last Order>
Order Display
<Search>

1.2 Definition of Terms

1.2.1 The term **N unique IDs** is used in this specification to refer to a field that must be able to hold any one ID within a minimum set of N unique IDs, regardless of the physical representation (e.g., binary, packed decimal, alphabetic, etc.) of the field.

1.2.2 The term **Fixed text, size N** is used in this specification to refer to a field that must be able to hold any string of characters of a fixed length of N. If the string it holds is shorter than N characters, it must be padded with spaces.

1.2.3 The term **Variable text, size N** is used in this specification to refer to a field that must be able to hold any string of characters of a variable length with a maximum length of N. The field may optionally be implemented as "fixed text, size N".

1.2.4 The term **Date and time** is used in this specification to refer to a field that must be able to hold any date between 1st January 1900 and 31st December 2100 with a resolution of at least one second.

1.2.5 The term **Numeric, N digits** is used in this specification to refer to a field that must be able to hold any N decimal digits value. Numeric fields that contain monetary values must use data types that give exact representation to at least the smallest monetary unit in the currency being used. For example, C_BALANCE and C_YTD_PMT in U.S. dollars may be represented as (12,2) digit signed decimal (with implicit scaling), or scaled to cents in a signed integer of at least 41 bits, or scaled to cents in a double precision (64 bit) REAL.

1.2.6 The term **Null** is used in this specification to mean out of the range of valid values for a given datatype and always the same value.

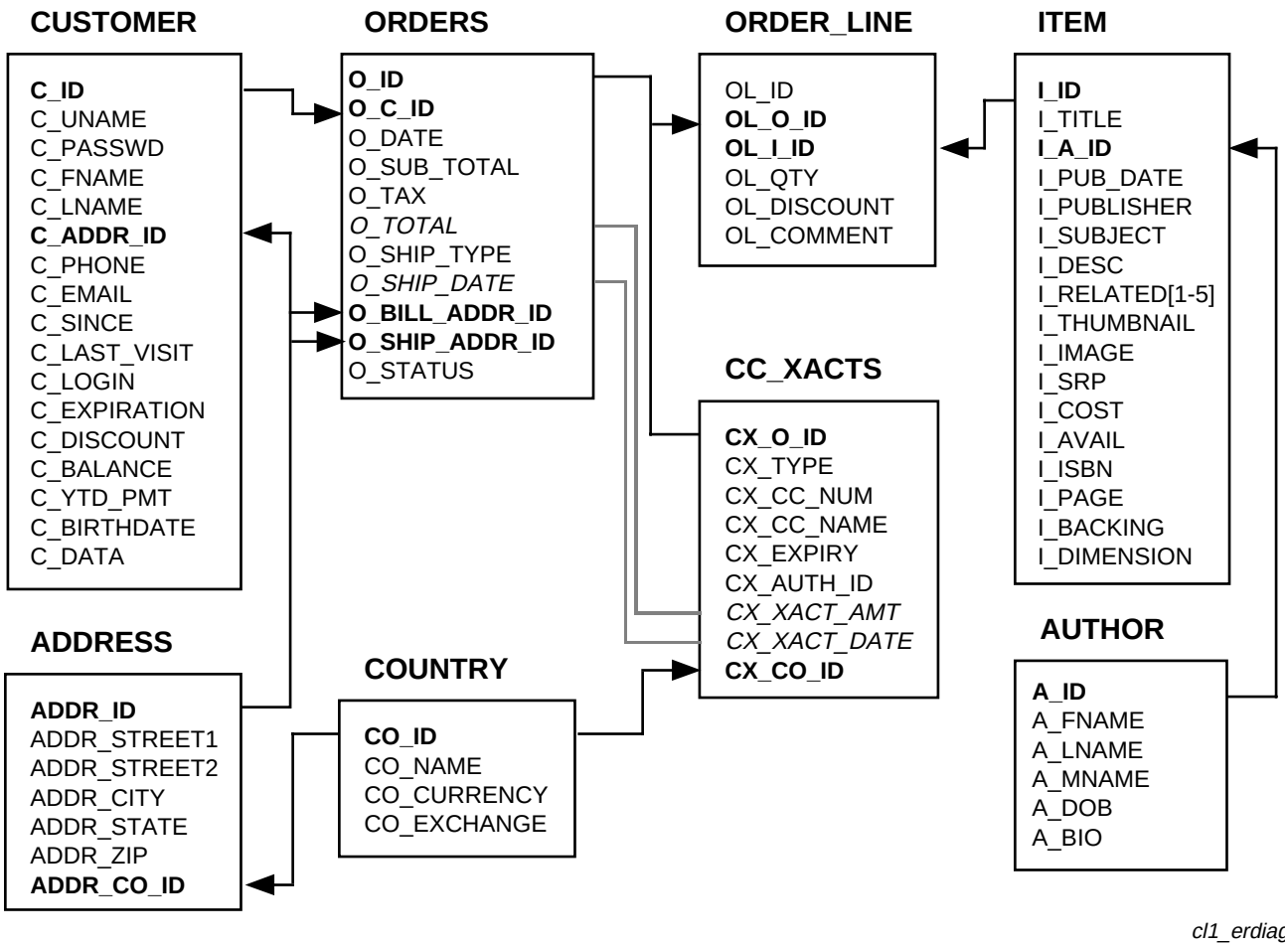
1.2.7 The term **Image** is used in this specification to refer to an implementation specific method of storing an image or a pointer to an image stored elsewhere on the system under test.

1.2.8 The term **application program** is used in this specification to refer to code that is not part of the commercially available components of the system, but produced specifically to implement the web interactions and the database transactions defined in this benchmark. For example, stored procedures, triggers, and referential integrity constraints are considered part of the application program when used to implement any portion of the web interactions or database transactions, but are not considered part of the application program when solely used to enforce integrity rules (see Clause 1.7) or transparency requirements (see Clause 1.8) independently of any specific web interaction or database transaction.

1.3 Database Entities, Relationships, and Characteristics

1.3.1 The components of the TPC-W database are defined to consist of a minimum of eight separate and individual tables. The relationships among these tables are defined in the entity-relationship diagram shown below and are subject to the rules specified in Clause 1.5.

Comment: To enable commercial products (commerce or merchant applications) to execute the workload without extensive modifications, a superset of the database schema is allowed. This could either be in the form of additional tables or additional fields in the tables. All such additions and/or modifications should be fully disclosed.



Legend:

- Doted lines represent one-to-one relationships between non-key fields related through a business rule. These fields are shown in italic.
- The arrows point in the direction of one-to-many relationships between tables.
- Bold types identify primary and foreign keys.

1.4 Table Layouts

The following lists define the minimal structure (list of fields) of each table:

Comment 1: For each table, the following fields can be implemented in any order, using any physical representation available from the tested system.

Comment 2: Table and column names are used for illustration purposes only; different names may be used by the implementation.

1.4.1 ITEM Table Layout

Field Name	Field Definition	Comments
I_ID	Numeric, 10 digits	Unique ID of Item
I_TITLE	Variable text, size 60	Title of Item
I_A_ID	Numeric, 10 digits	Author ID of Item
I_PUB_DATE	Date	Date of release of the product
I_PUBLISHER	Variable text, size 60	Publisher of item
I_SUBJECT	Variable text, size 60	Subject of Book
I_DESC	Variable text, size 500	Description of Item
I_RELATED1	Numeric, 10 digits	Unique Item ID (I_ID)
I_RELATED2	Numeric, 10 digits	Unique Item ID (I_ID)
I_RELATED3	Numeric, 10 digits	Unique Item ID (I_ID)
I_RELATED4	Numeric, 10 digits	Unique Item ID (I_ID)
I_RELATED5	Numeric, 10 digits	Unique Item ID (I_ID)
I_THUMBNAIL	Image	Thumbnail image of Item or pointer to thumbnail image
I_IMAGE	Image	Item image or pointer to image
I_SRP	Numeric, (15,2) digits	Suggested Retail Price
I_COST	Numeric, (15,2) digits	Cost of Item
I_AVAIL	Date	When item is available
I_ISBN	Fixed text, size 13	Product ISBN
I_PAGE	Numeric, 4 digits	Number of pages of book
I_BACKING	Variable text, size 15	Type of book, paper or hard back
I_DIMENSIONS	Variable text, size 25	Size of book in inches
Primary Key: (I_ID) (I_A_ID) Foreign Key, references (A_ID)		

1.4.2 COUNTRY Table Layout

Field Name	Field Definition	Comments
CO_ID	Numeric, 4 digits	Unique Country ID
CO_NAME	Variable text, size 50	Name of Country
CO_EXCHANGE	Numeric, (8,2) digits	Exchange rate to US Dollars
CO_CURRENCY	Variable text, size 10	Name of Currency
Primary Key: (CO_ID)		

1.4.3 AUTHOR Table Layout

Field Name	Field Definition	Comments
A_ID	Numeric, 10 digits	Unique Author ID
A_FNAME	Variable text, size 20	First Name of Author
A_LNAME	Variable text, size 20	Last Name of Author
A_MNAME	Variable text, size 20	Middle Name of Author
A_DOB	Date	Date of Birth of Author
A_BIO	Variable text, size 500	About the Author
Primary Key: (A_ID)		

1.4.4 CUSTOMER Table Layout

Field Name	Field Definition	Comments
C_ID	Numeric, 10 digits	Unique ID per Customer
C_UNAME	Variable text, size 23	Unique User Name for Customer
C_PASSWD	Variable text, size 14	User Password for Customer
C_FNAME	Variable text, size 15	First name of customer
C_LNAME	Variable text, size 15	Last name of customer
C_ADDR_ID	Numeric, 10 digits	Address ID of Customer
C_PHONE	Variable text, size 16	Phone number of customer
C_EMAIL	Variable text, size 50	For sending purchase confirmations
C_SINCE	Date	Customer member since
C_LAST_VISIT	Date	Time of Last Visit
C_LOGIN	Date and time	Start of Current Customer Session
C_EXPIRATION	Date and time	Current Customer Session Expiry
C_DISCOUNT	Numeric, (3,2) digits	Percentage discount for customer
C_BALANCE	Sign numeric, (15,2) digits	Balance of customer
C_YTD_PMT	Numeric, (15,2) digits	YTD Payment of customer
C_BIRTHDATE	Date	In case forget password
C_DATA	Variable text, size 500	Miscellaneous information
Primary Key: (C_ ID) (C_ADDR_ID) Foreign Key, references (ADDR_ID)		

1.4.5 ORDERS Table Layout

Field Name	Field Definition	Comments
O_ID	Numeric, 10 digits	Unique ID per order
O_C_ID	Numeric, 10 digits	Customer ID of Order
O_DATE	Date and time	Order Date
O_SUB_TOTAL	Numeric, (15,2) digits	Subtotal of all order-line items
O_TAX	Numeric, (15,2) digits	Tax over the subtotal
O_TOTAL	Numeric, (15,2) digits	Total for this order
O_SHIP_TYPE	Variable text, size 10	Method of delivery
O_SHIP_DATE	Date and time	Order Ship Date
O_BILL_ADDR_ID	Numeric, 10 digits	Address ID to bill
O_SHIP_ADDR_ID	Numeric, 10 digits	Address ID to ship order
O_STATUS	Variable text, size 15	Order status

Primary Key: (O_ID)
(O_C_ID) Foreign Key, references (C_ID); (O_BILL_ADDR, O_SHIP_ADDR) Foreign Key, references (ADDR_ID)

1.4.6 ORDER_LINE Table Layout

Field Name	Field Definition	Comments
OL_ID	numeric, 3 digits	Unique Order Line Item ID
OL_O_ID	numeric, 10 digits	Order ID of Order Line
OL_I_ID	numeric, 10 digits	Unique Item ID (I_ID)
OL_QTY	numeric, 3 digits	Quantity of Item
OL_DISCOUNT	numeric, (3,2) digits	Percentage discount off of I_SRP
OL_COMMENTS	variable text, size 100	Special Instructions

Primary Key: (OL_ID, OL_O_ID)
(OL_I_ID) Foreign Key, references (I_ID); (OL_O_ID) Foreign Key, references (O_ID)

1.4.7 CC_XACTS Table Layout

Field Name	Field Definition	Comments
CX_O_ID	Numeric, 10 digits	Unique Order ID (O_ID)
CX_TYPE	Variable text, size 10	Credit card type
CX_NUM	Numeric, 16 digits	Credit card number
CX_NAME	Variable text, size 30	Name on credit card
CX_EXPIRY	Date	Expiration date of credit card
CX_AUTH_ID	Fixed text, size 15	Authorization for transaction amount
CX_XACT_AMT	Numeric, (15,2) digits	Amount for this transaction
CX_XACT_DATE	Date and time	Date of authorization
CX_CO_ID	Numeric, 4 digits	Country where transaction originated

Primary Key: (CX_O_ID)
(CX_O_ID) Foreign Key, references (O_ID); (CX_CO_ID) Foreign Key, references (CO_ID)

1.4.8 ADDRESS Table Layout

Field Name	Field Definition	Comments
ADDR_ID	Numeric, 10 digits	Unique address ID
ADDR_STREET1	Variable text, size 40	Street address, line 1
ADDR_STREET2	Variable text, size 40	Street address, line 2
ADDR_CITY	Variable text, size 30	Name of city
ADDR_STATE	Variable text, size 20	Name of state
ADDR_ZIP	Variable text, size 10	Zip code or Postal code
ADDR_CO_ID	Numeric, 4 digits	Unique ID of Country

Primary Key: (ADDR_ID)
(ADDR_CO_ID) Foreign Key, references (CO_ID)

1.5 Web Objects, Characteristics and Relationship with Database.

1.5.1 The components of the TPC-W web objects are defined to consist of images of varying sizes in JPEG format. The currently defined sizes are: 5K, 10K, 50K, 100K and 250KB. The objects are meant to represent data flow of different media on the web such as images, audio, video-stream, animated GIF, etc. Each of these images is generated using the TPC-W Image Generator. The total number or images generated is equal to the cardinality of the ITEM table to represent each item's image and other rich media on the site (see Clause 4.6). The distribution for the different image sizes are as follows:

- 5K images 45%
- 10K images 35%

- 50K images 15%
- 100K images 4%
- 250K images 1%

1.5.2 Each item from the ITEM table also has a thumbnail image, I_THUMBNAIL, of 5 K-bytes associated with it. This image is generated using the TPC-W Image Generator.

1.5.3 The store-front has predetermined GIF images for navigation. This includes the buttons such as those for <Home>, <Search>, <Shopping Cart>, and <Checkout>. The implementation of the benchmark must use the button images that are part of the TPC-W sample web pages.

1.6 Implementation Rules

1.6.1 The physical clustering of records within the database is allowed.

1.6.2 A view which represents the rows to avoid logical read / writes is excluded.

Comment: The intent of this clause is to insure that the application implements the number of logical operations defined in the transaction profiles without combining several operations in one, via the use of a view.

1.6.3 All tables must have the properly scaled number of rows as defined by the database population requirements (see Clause 4.3).

1.6.4 Horizontal partitioning of tables is allowed. Groups of rows from a table may be assigned to different files, disks, or areas. If implemented, the details of such partitioning must be disclosed.

1.6.5 Vertical partitioning of tables is allowed. Groups of fields (columns) of one table may be assigned to files, disks, or areas different from those storing the other fields of that table. If implemented, the details of such partitioning must be disclosed (see Clause 1.6 for limitations).

Comment: In the two clauses above (1.5.4 and 1.5.5) assignment of data to different files, disks, or areas not based on knowledge of the logical structure of the data (e.g., knowledge of row or attribute boundaries) is not considered partitioning. For example, distribution or striping over multiple disks of a physical file which stores one or more logical tables is not considered partitioning as long as this distribution is done by the hardware or the operating system without knowledge of the logical structure stored in the physical file.

1.6.6 Replication is allowed for all tables. Manipulation of data in all copies of tables which are replicated must meet all requirements for atomicity, consistency, and isolation as defined in Clause 3. If implemented, the details of such replication must be disclosed.

Comment: Only one copy of a replicated table needs to meet the durability requirements defined in Clause 3.

1.6.7 Fields may be added and/or duplicated from one table to another as long as these changes do not improve performance.

1.6.8 Each field, as described in Clause 1.4, must be logically discrete and independently accessible by the data manager. For example, A_FNAME, A_MNAME and A_LNAME cannot be implemented as three sub-parts of a discrete attribute A_NAME.

1.6.9 Each field, as described in Clause 1.4, must be accessible by the data manager as a single field. For example, C_DATA cannot be implemented as two discrete fields C_DATA_1 and C_DATA_2.

Comment: The following fields are exceptions to this clause: All fields holding a time-and-date value (i.e., C_LOGIN, C_EXPIRATION, O_DATE, O_SHIP_DATE and CX_XACT_DATE) can be implemented as a combination of two fields: a date field and a time field. No vertical partitioning can be defined between the two fields used to implement it.

1.6.10 The primary key of each table must not directly represent the physical disk addresses of the row or any offsets thereof. The application may not reference rows using relative addressing since they are simply offsets from the beginning of the storage space. This does not preclude hashing schemes or other file organizations which have provisions for adding, deleting, and modifying records in the ordinary course of processing.

Comment 1: It is the intent of this clause that the application program (see Clause 1.2.8) executing the database transaction, or submitting the database transaction request, not use physical identifiers, but logical identifiers for all accesses, and contain no user written code which translates or aids in the translation of a logical key to the location within the table of the associated row or rows. For example, it is not legitimate for the application to build a "translation table" of logical-to-physical addresses and use it to enhance performance.

Comment 2: Internal record or row identifiers, for example, tuple IDs or cursors, may be used under the condition that within each database transaction, initial access to any row must be via a logical key comprised only of fields from that row. Initial access includes insertion, deletion, retrieval, and update of any row.

1.6.11 In the case where web objects are stored outside of the ITEM table and referenced in the ITEM table by a pointer managed by the application program (see Clause 1.2.8), that pointer must not directly represent the physical disk addresses of the object or any offsets thereof. The application may not reference objects using relative addressing since they are simply offsets from the beginning of the storage space. This does not preclude hashing schemes or other file organizations which have provisions for adding, deleting, and modifying objects in the ordinary course of processing.

1.6.12 While inserts and deletes are not performed on all tables, the system must not be configured to take special advantage of this fact during the test. Although inserts are inherently limited by the storage space available on the configured system, there must be no restriction on inserting in any of the tables a minimum number of rows equal to 5% of the table cardinality and with a key value of at least double the range of key values present in that table.

Comment: It is required that the space for the additional 5% table cardinality be configured for the Test Run and priced accordingly. If a commercial product is used for the application which requires a superset of the database schema, then the space configured and priced should include the additional storage needed for the additional tables and/or fields. For systems where space is configured and dynamically allocated at a later time, this space must be considered as allocated and included in the priced system (see Clause 4.4).

1.6.13 The minimum decimal precision for any computation performed as part of the application program (see Clause 1.2.8) must be the maximum decimal precision of all the individual items in that calculation.

1.6.14 Any other rules specified elsewhere in this document apply to the implementation (e.g., the consistency rules in Clause 3.3).

1.7 Integrity Rules

1.7.1 In any committed state, the primary key values must be unique within each table. For example, in the case of a horizontally partitioned table, primary key values of rows across all partitions must be unique.

1.7.2 In any committed state, no ill-formed rows may exist in the database. An ill-formed row occurs when the value of any attributes cannot be determined. For example, in the case of a vertically partitioned table, a row must exist in all the partitions.

1.8 Data Access Transparency Requirements

Data Access Transparency is the property of the system which removes from the application program any knowledge of the location and access mechanisms of partitioned data. An implementation which uses vertical and/or horizontal partitioning must meet the requirements for transparent data access described here.

No finite series of tests can prove that the system supports complete data access transparency. The requirements below describe the minimum capabilities needed to establish that the system provides transparent data access.

Comment: The intent of this clause is to require that access to physically and/or logically partitioned data be provided directly and transparently by services implemented by commercially available layers below the application program such as the data/file manager (DBMS), the operating system, the hardware, or any combination of these.

1.8.1 Each of the eight tables described in Clause 1.4 must be identifiable by names which have no relationship to the partitioning of tables. All data manipulation operations in the application program (see Clause 1.2.8) must use only these names.

1.8.2 The system must prevent any data manipulation operation performed using the names described in Clause 1.8.1 which would result in a violation of the integrity rules (see Clause 1.6). For example: the system must prevent a non-TPC-W application from committing the insertion of a row in a vertically partitioned table unless all partitions of that row have been inserted.

1.8.3 Using the names which satisfy Clause 1.8.1, any arbitrary non-TPC-W application must be able to manipulate any set of rows or columns:

- Identifiable by any arbitrary condition supported by the underlying DBMS
- Using the names described in Clause 1.8.1 and using the same data manipulation semantics and syntax for all tables.

For example, the semantics and syntax used to update an arbitrary set of rows in any one table must also be usable when updating another arbitrary set of rows in any other table.

Comment: The intent is that the TPC-W application program uses general purpose mechanisms to manipulate data in the database.

Clause 2 - WEB INTERACTIONS AND WORKLOAD PROFILE

2.1 Definition of Terms

2.1.1 The term **User** is used in this specification to refer to an entity (usually a human; could be a software program) that communicates with the SUT via a Browser.

2.1.2 The term **Browser** as used in this specification refers to the interface device capable of entering and displaying HTTP V 1.0, HTTP V1.1, HTML V 3.0, JavaScript, Java, Images (GIF & JPEG), Audio and Movie streams via TCP/IP. A browser is defined as the components that facilitate User input and the display of the output pages as defined in Clause 2.

2.1.3 The term **Emulated Browser** (**EB**) is used in this specification to refer to the entity (e.g., a process or a thread) that emulates a user communicating via a Browser by sending and receiving HTML content via HTTP and TCP/IP over a network connection (e.g., a socket) to the SUT.

2.1.4 The term **web interaction** is used in this specification to refer to a complete cycle of communication between the EB and the SUT. This cycle starts when the EB selects a navigation option from the previously displayed web page or when requesting the Home Page for the first time. It includes one or more exchanges of messages between the SUT and the EB. These exchanges may include the request and communication of cookies, HTML pages, picture files or other web objects. The number and types of such exchanges are benchmark implementation specific. The cycle also includes some processing to take place on the SUT. The cycle is completed when the last byte of data from the response page, including all referenced embedded objects, has been received by the EB.

2.1.5 The term **User Session** is used in this specification to refer to a continuous period of time during which an EB requests one or more web interactions, starting with the Home web interaction.

2.1.6 The term **Customer** is used in this specification to refer to a User whose identity has been recorded by the SUT in the CUSTOMER table.

2.1.7 The term **Shopping Session** is used in this specification to refer to a continuous period of time during which an EB requests one or more web interactions, starting as early as when it first requests the Home web interaction and no later than when it requests the Shopping Cart web interaction for the first time. The Shopping Session must be maintained by the SUT for at least two hours after the last update to the CART (during either the Shopping Cart or the Buy Confirm web interactions.)

Comment: The information required to be able to identify if the User requesting the web interaction has an active Shopping Session must be communicated by the RBE to the SUT during each web interaction (see Clause 2.2.5).

2.1.8 The term **Web Logging** is used in this specification to refer to entries in the Web Server Access Log.

2.1.9 The term **database transaction** as used is this specification refers to one or more operations that result in a unit of work on the database with full ACID properties as described in Clause 3. A web interaction may be comprised of one or more database transactions. When a database transaction includes more than one operations, the set of operations is enclosed between the tags <start transaction> and <end transaction>.

2.1.10 The term **atomic set of operations** is used in this specification to refer to a set of operations that results in data updates (addition, deletion, modification) within the SUT and has the property to guarantee that either all updates will be completed; or that none of the updates will be completed, leaving all targeted data unchanged. When defined as atomic, a set of operations is enclosed between the tags <start atomic set> and <end atomic set>.

2.1.11 The term **obtained** is used in this specification to refer to the action of obtaining the current value of a given field from within the SUT. The location within the SUT from which the value is obtained, such as database, cache, or other, is not specified and is only constrained by the web interaction definitions and by the ACID requirements.

2.1.12 The term **randomly selected within [x .. y]** means independently selected at random and uniformly distributed between x and y, inclusively, with a mean of (x+y)/2, and with the same number of digits of precision as shown. For example, [0.01 .. 100.00] has 10,000 unique values, whereas [1 ..100] has only 100 unique values.

2.1.13 The term **non-uniform random function (NURand)** is used in this specification to refer to the function used for generating C_ID and the targets for the Search Request web interactions. This function generates an independently selected and non-uniformly distributed random number over the specified range of values [*x* .. *y*], and is specified as follows:

NURand(A, x, y) = ((random(0, A) | random(x, y)) % (y - x + 1)) + x

Where:

expr1 | expr2 stands for the bitwise logical OR operation between expr1 and expr2

expr1 % expr2 stands for expr1 modulo expr2

random(x, y) stands for randomly selected within [*x* .. *y*]

A is a constant chosen according to the size of the range [x .. y]

2.2 Implementation Rules

2.2.1 A web interaction can be either unprotected or secure. In secure web interactions all communications between the EB and the SUT must be encrypted and communicated with SSL version 3, or above, using a minimum key of 40 bits and a DES cipher. The private key, for the server digital certificate must be at least 512 bits.

2.2.2 Each User Session must establish its own SSL session and execute a full handshake. SSL sessions must not be shared by multiple User Sessions.

2.2.3 When starting a new User Session, the EB must emulate one of the following two scenarios:

1. Submit a request from a User that is not a Customer. At some point, the User may become a Customer. This scenario is chosen by the EB 20% of the time.
2. Submit a request from a Customer and provide its Customer identification to the SUT. This scenario is chosen by the EB 80% of the time. The Customer identification (C_ID) of the returning Customer is chosen as specified in clause 2.3.2.

Comment1: Submitting a request from a Customer who does not provide its Customer identification to the SUT but, at some later time, identifies itself as a Customer is a scenario that is excluded from the benchmark. This scenario, while realistic, would increase the level of complexity without adding significant value to the workload.

Comment2: It is possible for a User to perform a User Session without ever identifying itself as a known Customer or without ever becoming a new Customer.

2.2.4 All Customers must identify themselves at the beginning of their User Session by communicating their C_ID, or any other unique identifier allowing the SUT to obtain their C_ID, during the first Home web interaction.

Comment: Communicating a Customer identification to the SUT can be implemented by various methods including, but not limited to, the passing of a field in the HTTP request, or the use of a cookie. The unique customer identifier is the only information about the Customer's identity that can be communicated by the Customer for the purpose of identification. There are no additional restrictions placed on format and content of the HTTP field or of the cookie or on other mechanism used for that purpose.

2.2.5 The SUT may choose to track all User Sessions using a unique SHOPPING_ID. Once a User Session becomes a Shopping Session, the SUT must track it using a unique SHOPPING_ID and an associated CART.

2.2.6 For each Shopping Session, the SUT must maintain an individual CART data structure that represents the User's shopping cart. This CART is associated with the unique SHOPPING_ID assigned by the SUT to the Shopping Session. At minimum, the CART must be able to maintain the following information:

- SC_SHOPPING_ID — Unique identifier of the Shopping Session
- SC_C_ID — The C_ID of the Customer
- SC_DATE — The date and time when the CART was created
- SC_SUB_TOTAL — The gross total amount of all items in the CART

- SC_TAX — The tax based on the gross total amount
- SC_SHIP_COST — The total shipping and handling charges
- SC_TOTAL — The total amount of the order
- SC_C_FNAME — C_FNAME of the Customer
- SC_C_LNAME — C_LNAME of the Customer
- SC_C_DISCOUNT — C_DISCOUNT of the Customer
- A minimum of 100 items with:
  - SCL_I_ID — The I_ID for the item in the CART
  - SCL_QTY — The quantity in the CART
  - SCL_COST — The cost of the item in the CART
  - SCL_SRP — The list price for the item in the CART
  - SCL_TITLE — The title of the item in the CART
  - SCL_BACKING — The backing of the item in the CART

The physical implementation of the CART is not specified as long as it supports the above logical representation, that it is maintained by the SUT through the duration of the Shopping Session and that it is kept durable over any single point of failure (see Clause 3).

2.2.7 The SUT may choose to track all Customers using their C_ID. Once a Customer has been identified as the owner of a Shopping Session, its identity must be maintained within the CART associated with that Shopping Session.

2.2.8 The clauses titled **Input Requirements** define the minimum set of data required by the SUT as input to a web interaction. Similarly, the clause titled Navigation Options define the minimum set of data to be provided by the EB as input to a web interaction. The implementation may choose to augment these minimum sets of input data provided that it does not reduce the workload on the SUT and does not improve performance.

2.2.9 The clauses titled **Pseudo-code Example** give an example of how the processing requirements of the web interaction could be implemented. This pseudo-code is for example only and does not take precedence over the requirement defined in the clauses titled Processing Definition.

2.2.10 The clauses titled **Response Page Definition** include a sample of each response page and a set of minimum requirements for the HTML code produced by the SUT for that response page. In addition to these page specific requirements, all response pages must meet the following requirements:

1. The page content as displayed in the browser must be the same as that of the sample response page.
2. The page layout as displayed in the browser must be similar to that of the sample response page.
3. All elements of the page (i.e., web objects) must be communicated individually.
4. All tags in the HTML code must be W3C compliant.
5. The HTML code may not contain or reference any active elements.

2.2.11 If caching is used, it must meet all the requirements of Clause 6.2.3.

2.2.12 The clauses titled **EB Navigation Options** define the set of navigation options that the EB must choose from. An actual user has many additional navigation options (e.g., bookmarks, go menu, etc.) Those additional options (not included in the EB Navigation Options) are considered secondary and are not exercised during the benchmark execution.

2.2.13 The **Web Server Access Log** data must be collected with a minimum resolution of one second and written at least once every 30 seconds to a persistent media, and must contain all fields specified in the Common Log Format (CLF). These are as follows:

- client: DNS Name or IP address of remote client
- dd: Day
- mm: Month
- yyyy: Year
- hh: hour
- mm: minutes
- ss: seconds
- request: First line of HTTP request as sent by the client
- ddd: status code returned by the server
- bbbb: Total number of bytes sent, not including HTTP header

2.2.14 After an EB requests a New Products, Best Sellers, or Search Result web interaction, the EB must maintain a cached list of the URL's (**CURL**) that were contained in the response page for that web interaction, until such time that the EB selects a navigation option that is not the Product Detail web interaction. This CURL is maintained for the sole purpose of navigating from the Product Detail interaction back to the Product Detail interaction, as specified in clause 2.14.5.4.

2.2.15 The implementation of **flags**, as used in the web interaction definitions, is not specified. Flags are used to carry information from one web interaction to another. Examples of possible implementations include the use of a field in the HTTP request and the use of multiple URL's, each reflecting a different value for the flag.

2.2.16 There is no requirement for the implementation of the benchmark to allow the browser to input or to display any characters other than those defined in Clause 4.6.2.2.

2.2.17 The generic **Promotional Processing**, executed by selected web interactions, is defined as follows:

- The SUT selects a random I_ID from a uniform distribution over the range of values define in Clause 4.
- The SUT obtains the five related items (I_RELATED1, I_RELATED2, I_RELATED3, I_RELATED4, I_RELATED5) for the selected item.
- The SUT obtains the thumbnail image, I_THUMBNAIL, from the five related items.

2.3 Home Web Interaction

2.3.1 Overview

This unprotected web interaction returns to the EB a web page which contains links to product lists for new products and for best sellers. This is the initial web interaction requested by all Users starting a new User Session. It is also a navigation option from most other web pages.

2.3.2 Input Requirements

This web interaction is invoked by an HTTP request and requires the following input data:

- The URL for the Home Page, which is contained in the HTTP request.
- If this is the first Home web interaction of a User Session and the User is a Customer, then the EB presents as input a C_ID that was chosen using the NURand function defined in Clause 2.1.12 with the following parameters:

  x = 1

  y = NUM_CUSTOMERS, where NUM_CUSTOMERS is the initial cardinality of the CUSTOMER table.

  A is chosen according to the following table:

For NUM_CUSTOMERS in this range	Value for A
1 - 9,999	1,023
10,000 - 39,999	4,095
40,000 - 159,999	16,383
160,000 - 639,999	65,535
640,000 - 2,559,999	262,143
2,560,000 - 10,239,999	1,048,575
10,240,000 - 40,959,999	4,194,303
40,960,000 - 163,839,999	16,777,215
163,840,000 - 655,359,999	67,108,863

Comment: This means that any C_IDs assigned to newly-created Customers during a Test Run (as specified in clause 2.6.3.2) are never used by RBEs for returning Customer selections. That is, a new Customer will never be a returning Customer. The SUT is not allowed to take advantage of this fact to improve its performance.

2.3.3 Processing Definition

2.3.3.1 If this User is a Customer and its C_ID is known, then the SUT obtains the following information about the Customer:

- C_FNAME

- C_LNAME

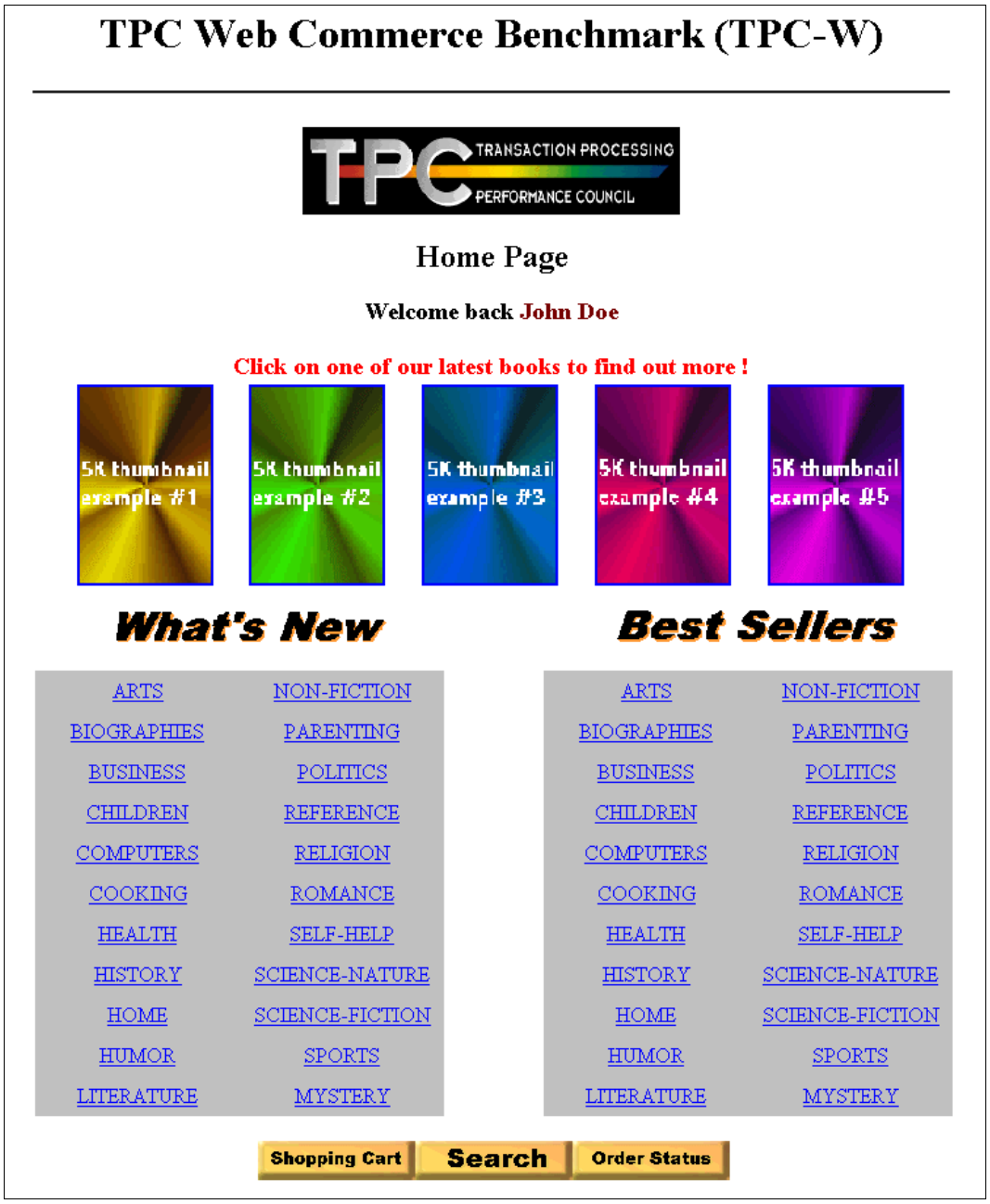
2.3.3.2 The SUT executes the Promotional Processing, as defined in Clause 2.2.17.

2.3.3.3 The SUT produces the HTML code for the Home Page containing a list of 24 subjects for new items, a list of 24 subjects for best sellers and, if a C_ID was known, the Customer's information; and returns the web page to the EB.

2.3.3.4 The EB selects a navigation option according to Clause 5.2.2.

2.3.4 Response Page Definition

A sample of the Home Page is shown below as it would appear in a Browser:



The HTML code for the Home Page (an example of which can be found in Appendix F) must meet the following requirements:

1. The length of the code must be a minimum of 6,900 characters (excluding spaces).
2. The following web objects must be referenced:
   - Images for the <Shopping Cart>, <Search> and <Order Status> buttons
   - Images for the TPC Logo
   - Images for the thumbnail of the five promotional items

2.3.5 EB Navigation Options

2.3.5.1 <Shopping Cart> button: Make an HTTP request to initiate the Shopping Cart Update web interaction, with the following input:

- An empty list of (I_ID, QTY) pairs
- The ADD_FLAG is set to "N"

2.3.5.2 < Search> button: Make an HTTP request to initiate the Search Request web interaction.

2.3.5.3 <Order Status> button: Make an HTTP request to initiate the Order Inquiry web interaction.

2.3.5.4 <New Products> link: Make an HTTP request to initiate the New Products web interaction, with the following input:

- SUBJECT_STRING, where the EB selects from a uniform random distribution one entry from the list of URLs for new title subjects.

2.3.5.5 <Best Sellers> link: Make an HTTP request to initiate the Best Sellers web interaction, with the following input:

- SUBJECT_STRING, where the EB selects from a uniform random distribution one entry from the list of URLs for best seller subjects.

2.4 Shopping Cart Web Interaction

2.4.1 Overview

This unprotected web interaction updates the associated CART (refreshing the CART's date and optionally adding a new item or updating existing items) and always returns to the EB a web page which displays the updated contents of the User's CART. If a Shopping Session is not identified at the start of this web interaction, a new Shopping Session is created.

2.4.2 Input Requirements

This web interaction is invoked by an HTTP request and requires the following input data:

- The URL for the Shopping Cart Page, which is contained in the HTTP request.
- A list of (I_ID, QTY) pairs for the items to be updated or added to the CART, which is contained in the HTTP request. This list may be empty.
- A value for the flag ADD_FLAG, which depends on the page from which this interaction is requested.

2.4.3 Processing Definition

2.4.3.1 If the SHOPPING_ID is not known, then the SUT creates a new unique SHOPPING_ID. If there is no CART associated with this SHOPPING_ID, then the SUT creates an associated CART initialized as follows:

- SC_SHOPPING_ID = SHOPPING_ID

- SC_DATE = current date and time on the SUT

2.4.3.2 If a new CART was created during this web interaction and C_ID is known, then the SUT obtains information about the Customer to update the CART as follows:

- SC_C_FNAME = C_FNAME
- SC_C_LNAME = C_LNAME
- SC_C_DISCOUNT = C_DISCOUNT

2.4.3.3 Based on SHOPPING_ID, the SUT updates the associated CART as an atomic set of operations as follows:

<start atomic set>

- If ADD_FLAG = "Y" (and the optional CART limit of 100 items has not been reached):
  - If I_ID = SCL_I_ID (i.e., the item already exists in the CART):
    - SCL_QTY = SCL_QTY + 1 (i.e., increment quantity by 1 for SCL_I_ID)
  - Else (i.e., the item does not already exists in CART), add the item to the CART with:
    - SCL_I_ID = I_ID
    - SCL_QTY = 1
    - SCL_COST = I_COST
    - SCL_SRP = I_SRP
    - SCL_TITLE = I_TITLE
    - SCL_BACKING = I_BACKING

  Note: If the CART limit has been reached, the item is not added and the message "Shopping Cart is full, item not added" is included on the response page. The processing resumes at Clause 2.4.3.4.
- If ADD_FLAG = "N" and the list of (I_ID, QTY) pairs is not empty, the CART is updated as follows:
  - For each (I_ID, QTY) pair:
    - If QTY = 0 (i.e., the item must be removed from the CART):
      - Delete the entry from the CART for item SCL_I_ID
    - Else, update the item as follows:
      - SCL_QTY = QTY (i.e., update the quantity for item SCL_I_ID)
- Else (i.e., ADD_FLAG = "N" and the list of (I_ID, QTY) pairs is empty), add a promotional item to the CART as follows:
  - A random I_ID is selected by the SUT within the range of [1 .. NUM_ITEMS]
  - The SUT obtains the value of I_RELATED1 for the selected I_ID
  - The SUT adds the item to the CART as follows:
    - The SUT obtains the following information about the item I_RELATED1:
      - I_COST
      - I_SRP

- I_TITLE
- I_BACKING

- A new item is added to the CART with:
  - SCL_I_ID = I_RELATED1
  - SCL_QTY = 1
  - SCL_COST = I_COST
  - SCL_SRP = I_SRP
  - SCL_TITLE = I_TITLE
  - SCL_BACKING = I_BACKING

♦ The new sub-total of the CART is computed as: SC_SUB_TOTAL = sum(SCL_QTY * SCL_COST)

♦ The value of SC_DATE is set to the current date and time on the SUT, with a minimum precision of 1 second.

<end atomic set>

2.4.3.4 The SUT executes the Promotional Processing, as defined in Clause 2.2.17.

2.4.3.5 Based on SHOPPING_ID, the SUT obtains the following information about the associated CART:

♦ For the CART:
  - SC_DATE
  - SC_SUB_TOTAL
  - SC_C_FNAME
  - SC_C_LNAME

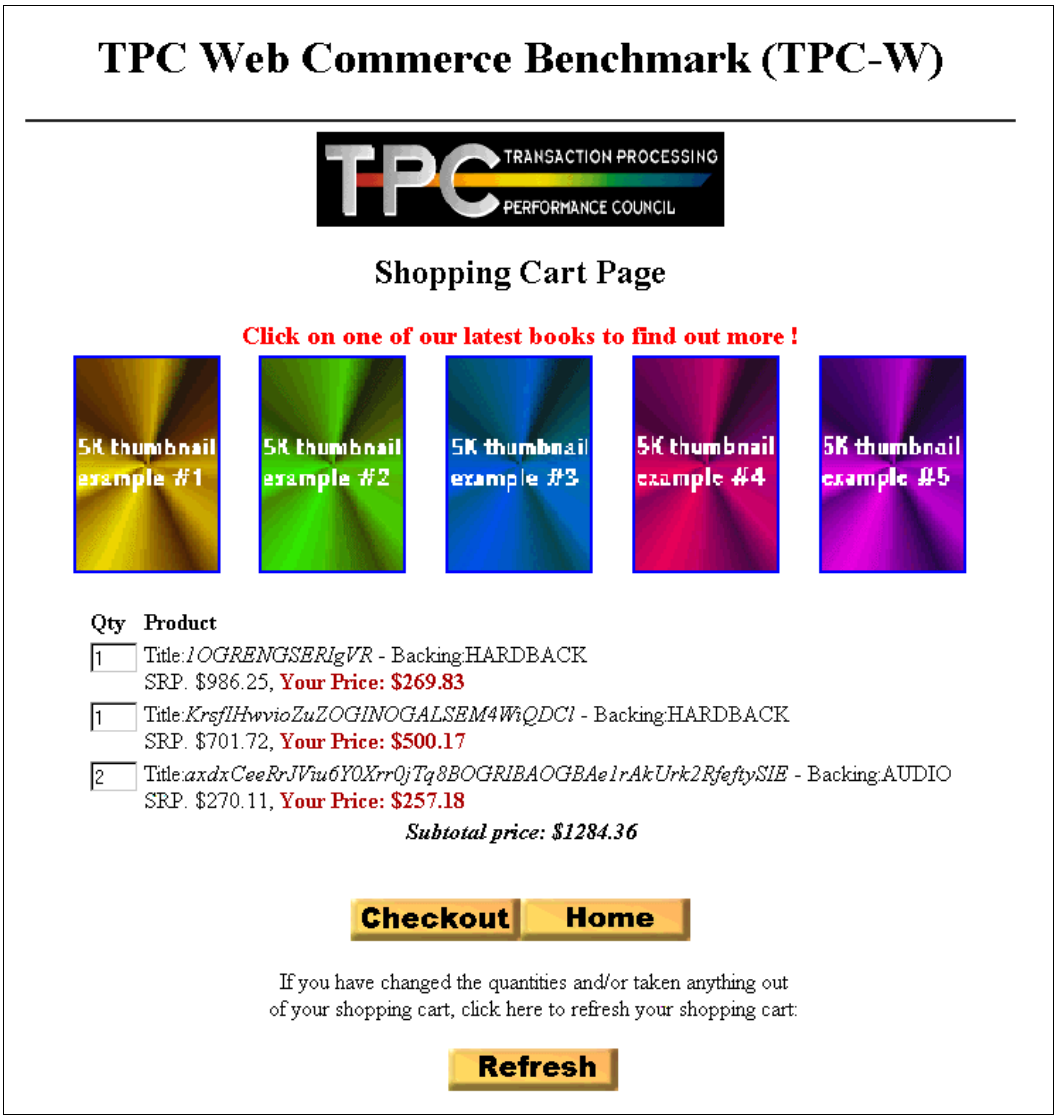
♦ For each item in the CART:
  - SCL_I_ID
  - SCL_QTY
  - SCL_COST
  - SCL_SRP
  - SCL_TITLE
  - SCL_BACKING

2.4.3.6 The SUT produces the HTML code for the Shopping Cart Page and returns the web page to the EB.

2.4.3.7 The EB selects a navigation option according to Clause 5.2.2.

2.4.4 Response Page Definition

A sample of the Shopping Cart Page is shown below as it would appear in a Browser:



The HTML code for the Shopping Cart Page (an example of which can be found in Appendix F) must meet the following requirements:

1. The length of the code must be a minimum of 2,300 characters (excluding spaces).
2. The following web objects must be referenced:
   - Images for the <Checkout>, <Home> and <Refresh> buttons
   - Images for the TPC Logo
   - Images for the thumbnail of the five promotional items

2.4.5 EB Navigation Options

2.4.5.1 <Refresh> button: Make an HTTP request to initiate the Shopping Cart web interaction, with the following input:

- The EB creates a complete list of (I_ID, QTY) pairs for all items in the CART shown in the most recent Shopping Cart Page received by the EB.

  Comment: The actions defined in the Processing Definition section of the Shopping Cart interaction guarantee that the CART will have at least one item in it, and therefore, the complete list of (I_ID, QTY) pairs created by the EB for this navigation option will also contain at least one pair.

- If there is only one (I_ID, QTY) pair in the list, the EB sets QTY to a random value (between 1 and 10, excluding SCL_QTY) generated using a uniform random function.
- Else there are greater than one (I_ID, QTY) pairs in the list. The EB selects a random number of (I_ID, QTY) pairs from the list (in the inclusive range [1, (one less than the number of pairs in the list)] using a uniform random function. For each selected pair, the EB sets QTY to a random value (between 0 and 10, excluding SCL_QTY) generated using a uniform random function.
- The ADD_FLAG is set to "N".

2.4.5.2 <Home> button: Make an HTTP request to initiate the Home web interaction.

2.4.5.3 <Checkout> button: Make an HTTP request to initiate the Customer Registration web interaction.

2.5 Customer Registration Web Interaction

2.5.1 Overview

This secure web interaction returns to the EB a web page which allows a User to provide the information necessary to register as a known Customer or as a new Customer and to submit their registration. This is the first step in buying the contents of the CART.

2.5.2 Input Requirements

This web interaction is invoked by an HTTP request and requires the following input data:

- The URL for the Customer Registration Page, which is contained in the HTTP request.

2.5.3 Processing Definition

2.5.3.1 If a C_ID is associated to the known SHOPPING_ID for this request, then the SUT obtains the following information about the Customer:

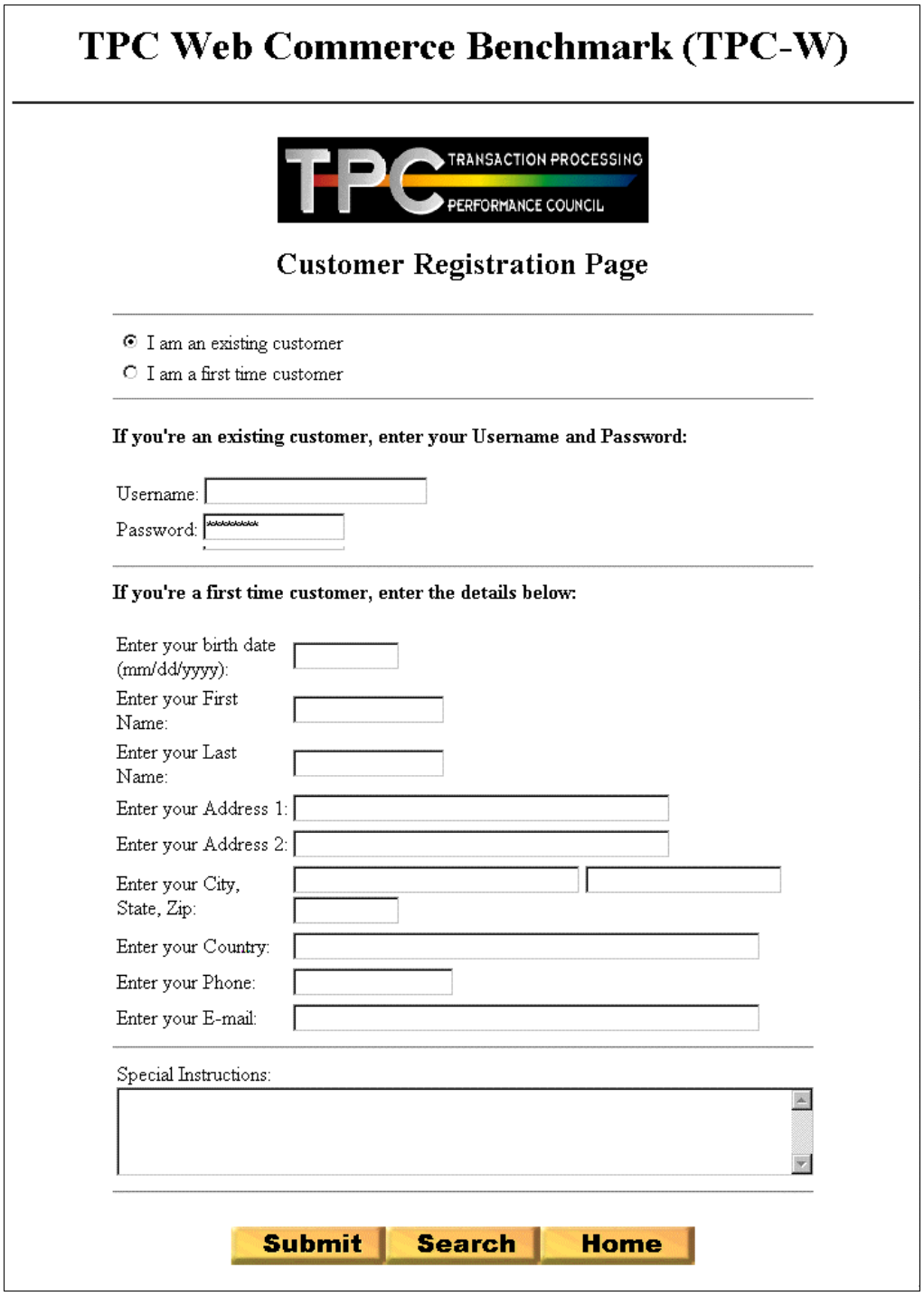
- C_FNAME
- C_LNAME

2.5.3.2 The SUT produces the HTML code for the Customer Registration Page and returns the web page to the EB.

2.5.3.3 The EB selects a navigation option according to Clause 5.2.2.

2.5.4 Response Page Definition

A sample of the Customer Registration Page is shown below as it would appear in a Browser:



The HTML code for the Customer Registration Page (an example of which can be found in Appendix F) must meet the following requirements:

1. The length of the code must be a minimum of 2,250 characters (excluding spaces).
2. The following web objects must be referenced:
   - Images for the <Submit>, <Home> and <Search> buttons

- Images for the TPC Logo

2.5.5 EB Navigation Options

2.5.5.1 <Submit> button with the following:

- If the EB knows the username (UNAME) associated with the current User Session, then the "Returning Customer" is checked: Make an HTTP request to initiate the Buy Request web interaction, with the following input:
  - RETURNING_FLAG set to "Y"
  - UNAME is generated by the EB according to Clause 4.6.2.9.
  - PASSWD is generated by the EB according to Clause 4.6.2.10.
- If the EB does not yet know a username (UNAME) associated with the current User Session, then the "Returning Customer" is un-checked: Make an HTTP request to initiate the Buy Request web interaction, with the following input:
  - RETURNING_FLAG is set to "N"
  - FNAME is a random a-string [8 .. 15]
  - LNAME is a random a-string [8 .. 15]
  - STREET1 is a random a-string [15 .. 40]
  - STREET2 is a random a-string [15 .. 40]
  - CITY is a random a-string [10 .. 30]
  - STATE is a random a-string [2 .. 20]
  - ZIP is a random a-string [5 .. 10]
  - COUNTRY is chosen according to Clause 4.6.2.14
  - PHONE is a random n-string [9 .. 16]
  - EMAIL is generated according to Clause 4.4.2.11
  - BIRTHDATE is a random date
  - DATA is a random a-string [100 .. 500]

2.5.5.2 <Home> button: Make an HTTP request to initiate the Home web interaction.

2.5.5.3 < Search> button: Make an HTTP request to initiate the Search Request web interaction.

2.6 Buy Request Web Interaction

2.6.1 Overview

This secure web interaction registers a new customer or identifies a returning customer and returns to the EB a web page which displays information about the customer, confirming either the registration or the identification, and displays a summary of the items in the associated CART. The page provides editable fields for entering credit card information and selecting shipping options.

2.6.2 Input Requirements

This web interaction is invoked by an HTTP request and requires the following input data:

- The URL for the Buy Request Page, which is contained in the HTTP request.
- Value for the field RETURNING_FLAG, which is contained in the HTTP request
- If RETURNING_FLAG is set to "Y", values for the following fields, which are contained in the HTTP request:
  - UNAME
  - PASSWD
- If RETURNING_FLAG is set to "N", values for the following fields, which are contained in the HTTP request:
  - FNAME
  - LNAME
  - STREET1
  - STREET2
  - CITY
  - STATE
  - ZIP
  - COUNTRY
  - PHONE
  - EMAIL
  - BIRTHDATE
  - DATA

2.6.3 Processing Definition

2.6.3.1 If RETURNING_FLAG is set to "Y", then the SUT executes the following steps:

- Obtains the following information about the customer with (C_UNAME = UNAME):
  - C_ID
  - C_PASSWD
  - C_FNAME
  - C_LNAME
  - ADDR_STREET1
  - ADDR_STREET2
  - ADDR_CITY
  - ADDR_STATE
  - ADDR_ZIP

- CO_NAME
- C_PHONE
- C_EMAIL
- C_BIRTHDATE
- C_DATA

♦ PASSWD is compared to C_PASSWD.

Comment: In a real-life application, an invalid password check would result in rejecting the purchase request. In the context of the benchmark, all supplied passwords will be valid, allowing the processing of the order to continue. The application program is not required to include any logic to deal with invalid passwords.

2.6.3.2 If RETURNING_FLAG is set to "N", the SUT executes the following steps within a single database transaction:

<start transaction>

♦ Adds a new customer record in the CUSTOMER table based on the input data and the following values:

- C_ID is set to a unique value, not necessarily serial or contiguous, within the CUSTOMER table.
- C_UNAME is generated according to Clause 4.6.2.9
- C_PASSWD is generated according to Clause 4.6.2.10
- C_LAST_VISIT is the current date/time given by the operating system when the record is added
- C_DISCOUNT is random within [0.00 .. 0.50]
- C_BALANCE is set to 0.00
- C_YTD_PMT is set to 0.00
- C_SINCE is the current date/time given by the operating system when the record is added
- C_FNAME is set to FNAME
- C_LNAME is set to LNAME
- C_PHONE is set to PHONE
- C_EMAIL is set to EMAIL
- C_BIRTHDATE is set to BIRTHDATE
- C_DATA is set to DATA

♦ Attempts to match the customer's address (STREET1, STREET2, CITY, STATE, ZIP, COUNTRY) with an address in the ADDRESS table. If a match is found, then the C_ADDR_ID is set to ADDR_ID for the matching record. If no match is found, a new record is created in the ADDRESS table using the customer's address with a new unique ADDR_ID (not necessarily sequential nor contiguous).

<end transaction>

2.6.3.3 The SUT executes the following steps as an atomic set of operations:

<start atomic set>

- Obtains the following information for each item in the CART:
  - SCL_TITLE
  - SCL_COST
  - SCL_SRP
  - SCL_BACKING
  - SCL_QTY
- Updates the following fields of the CART:
  - The value of SCL_COST is set to the current value of I_COST from the ITEM table for each item in the CART.
  - The value of SC_DATE is set to the current date and time, with a minimum precision of 1 second.
- Calculates and updates the following fields of the CART:
  - SC_SUB_TOTAL = sum(SCL_COST * SCL_QTY) * (1 – SC_C_DISCOUNT)
  - SC_TAX = SC_SUB_TOTAL * 0.0825
  - SC_SHIP_COST = 3.00 + (1.00 * sum(SCL_QTY))
  - SC_TOTAL = SC_SUB_TOTAL + SC_SHIP_COST + SC_TAX

<end atomic set>

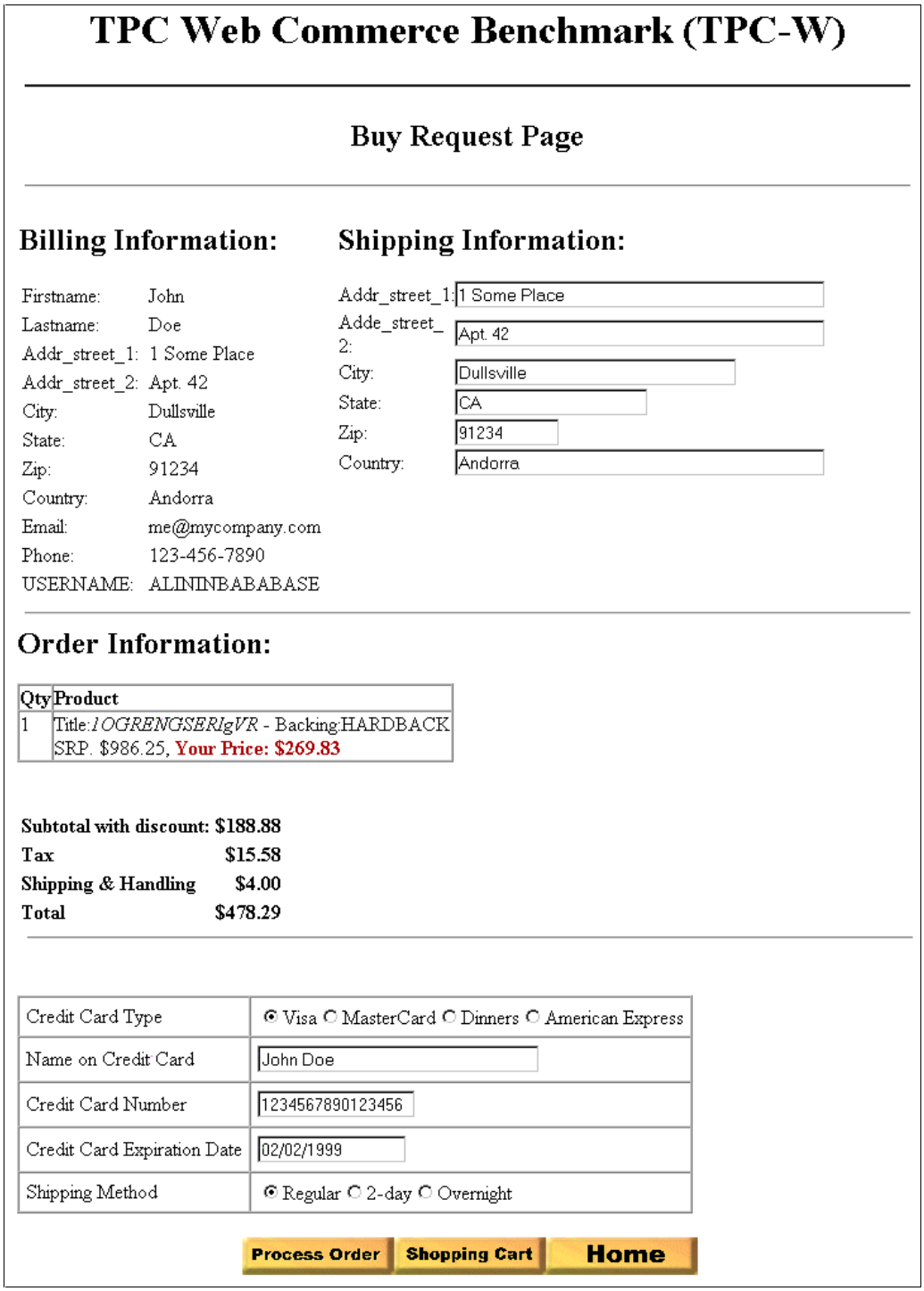
2.6.3.4 The SUT produces the HTML code for the Buy Request Page containing the customer information and the purchase information; and returns the web page to the EB. If RETURNING_FLAG is set to "N", then the SUT also returns to the EB the values of the C_ID and C_UNAME of the newly created CUSTOMER record.

Comment: Communicating the C_ID and C_UNAME values to the EB can be implemented by various methods including, but not limited to, the passing of a field in the HTML response, or the use of a cookie.

2.6.3.5 The EB selects a navigation option according to Clause 5.2.2.

2.6.4 Response Page Definition

A sample of the Buy Request Page is shown below as it would appear in a Browser:



TPC Web Commerce Benchmark (TPC-W)

Buy Request Page

Billing Information:

Firstname:	John
Lastname:	Doe
Addr_street_1:	1 Some Place
Addr_street_2:	Apt. 42
City:	Dullsville
State:	CA
Zip:	91234
Country:	Andorra
Email:	me@mycompany.com
Phone:	123-456-7890
USERNAME:	ALININBABABASE

Shipping Information:

Addr_street_1:	1 Some Place
Adde_street_2:	Apt. 42
City:	Dullsville
State:	CA
Zip:	91234
Country:	Andorra

Order Information:

Qty	Product
1	Title:*1OGRENGSERIgVR* - Backing:HARDBACK SRP. $986.25, **Your Price: $269.83**

Subtotal with discount:	**$188.88**
Tax	**$15.58**
Shipping & Handling	**$4.00**
Total	**$478.29**

Credit Card Type	◉ Visa ○ MasterCard ○ Dinners ○ American Express
Name on Credit Card	John Doe
Credit Card Number	1234567890123456
Credit Card Expiration Date	02/02/1999
Shipping Method	◉ Regular ○ 2-day ○ Overnight

Process Order | Shopping Cart | Home

The HTML code for the Buy Request Page (an example of which can be found in Appendix F) must meet the following requirements:

1. The length of the code must be a minimum of 3,600 characters (excluding spaces).
2. The following web objects must be referenced:
   - Images for the <Process Order>, <Home> and <Shopping Cart> buttons

2.6.5 EB Navigation Options

2.6.5.1 <Process Order> button: Make an HTTP request to initiate the Buy Confirm web interaction, with the following input:

- SHIPPING is randomly selected within [Regular, 2-day, Overnight]
- CC_TYPE is randomly selected within [VISA, MASTERCARD, DISCOVER, DINERS, AMEX]
- CC_NUMBER is generated as a random n-string of 16 digits
- CC_NAME is generated by concatenating C_FNAME and C_LNAME separated by a single space
- CC_EXPIRY is generated as the current date + a random number of days within [1 .. 730]
- On a randomly selected 5% of the time, the shipping address is updated with (STREET_1, STREET_2, CITY, STATE, ZIP, COUNTRY) generated according to Clause 4.7.1

2.6.5.2 <Home> button: Make an HTTP request to initiate the Home web interaction.

2.6.5.3 <Shopping Cart> button: Make an HTTP request to initiate the Shopping Cart web interaction.

- An empty list of (I_ID, QTY) pairs
- The ADD_FLAG is set to "N"

2.7 Buy Confirm Web Interaction

2.7.1 Overview

This secure web interaction transfers the content of the associated CART into a newly created order for the registered customer and executes a full payment authorization. It then returns to the EB a web page containing the details of the newly created order.

2.7.2 Input Requirements

2.7.2.1 This web interaction is invoked by an HTTP request and requires the following input data:

- The URL for the Buy Confirm Page, which is contained in the HTTP request.
- The following payment and shipping information:
  - CC_TYPE
  - CC_NUMBER
  - CC_NAME
  - CC_EXPIRY
  - SHIPPING
  - An optional shipping address (STREET, CITY, STATE, ZIP, COUNTRY)

2.7.3 Processing Definition

2.7.3.1 Based on the known SHOPPING_ID and C_ID, the SUT obtains information about the associated CART and Customer:

- The following information is obtained about each item in the CART:
  - SCL_I_ID
  - SCL_COST
  - SCL_QTY
- The following information is obtained about the CART:
  - SC_SUB_TOTAL
  - SC_TAX
  - SC_SHIP_COST
  - SC_TOTAL
- The following information is obtained about the Customer:
  - C_FNAME
  - C_LNAME
  - C_DISCOUNT
  - C_ADDR_ID

2.7.3.2 If a shipping address is passed, then the SUT attempts to match the address (STREET, CITY, STATE, ZIP, COUNTRY) with an address in the ADDRESS table. If no match is found, then a new record is created in the ADDRESS table using the shipping address with a new unique ADDR_ID table (not necessarily sequential or contiguous). This entire processing step is executed within a single database transaction.

2.7.3.3 The SUT executes the following steps as an atomic set of operations:

<start atomic set>

- Creates a new order as follows, within a single database transaction:

  <start transaction>

  - A record is added in the ORDER table with:
    - O_ID is unique within the ORDER table (not necessarily sequential or contiguous)
    - O_C_ID is set to C_ID
    - O_DATE is set to the current operating system date and time
    - O_SUB_TOTAL is set to SC_SUB_TOTAL
    - O_TAX is set to 8.25
    - O_TOTAL is set to SC_TOTAL
    - O_SHIP_TYPE is set to SHIPPING
    - O_SHIP_DATE is set to current operating system date + random within [1 .. 7] days
    - O_BILL_ADDR_ID is set to C_ADDR_ID
    - O_SHIP_ADDR_ID is set to either C_ADDR_ID or to the newly created ADDR_ID
    - O_STATUS is set to “Pending”
  - For each item in the CART, a record is added in the ORDER_LINE table with:

- OL_ID is unique within the ORDER_LINE record for the order (not necessarily sequential or contiguous)
- O_O_ID is set to O_ID
- OL_I_ID is set to SCL_I_ID
- OL_QTY is set to SCL_QTY
- OL_DISCOUNT is set to C_DISCOUNT
- OL_COMMENT is set to a random a-string [50 …100]

<end transaction>

♦ Obtain an authorization from the Payment Gateway Emulator (PGE) as follows:

- Build a string of bytes starting with CC_NUMBER enclosed by the tags <NUM> and </NUM>, followed by O_ID enclosed by the tags <OID> and </OID>; and padded to a total length L, where L is at least 6000 bytes, and the pad characters are a random a-string [L - 48 .. L - 48].
- Send the above string to the PGE, encrypted and communicated with SSL version 3, or above, using a minimum key of 40 bits and a DES cipher. The private key, for the digital certificate must be at least 512 bits.
- Wait to receive a response from the PGE (see Clause 6.3).

  Comment: An SSL handshake must be negotiated between the SUT and the PGE prior to the above exchange. The SSL session resulting from this handshake must not be used for more than 100 of these exchanges. The number of SSL sessions open at any given point in time between the SUT and the PGE is not limited.
- Decrypt the response from the PGE and extract the authorization code, AUTH_ID, from the message, which is enclosed by the tags <CODE> and </CODE>.

♦ Creates a new credit card record as follows, within a single database transaction:

<start transaction>

- A record is added to the CC_XACTS table with:
  - CX_O_ID is set to O_ID
  - CX_TYPE is set to CC_TYPE
  - CX_NUM is set to CC_NUMBER
  - CX_NAME is set to CC_NAME
  - CX_EXPIRY is set to CC_EXPIRY
  - CX_AUTH_ID is set to AUTH_ID
  - CX_XACT_AMT is set to SC_TOTAL
  - CX_XACT_DATE is set to the current date on the SUT
  - CX_CO_ID is set to COUNTRY from the shipping address

<end transaction>

♦ Clears all items from the CART.

<end atomic set>

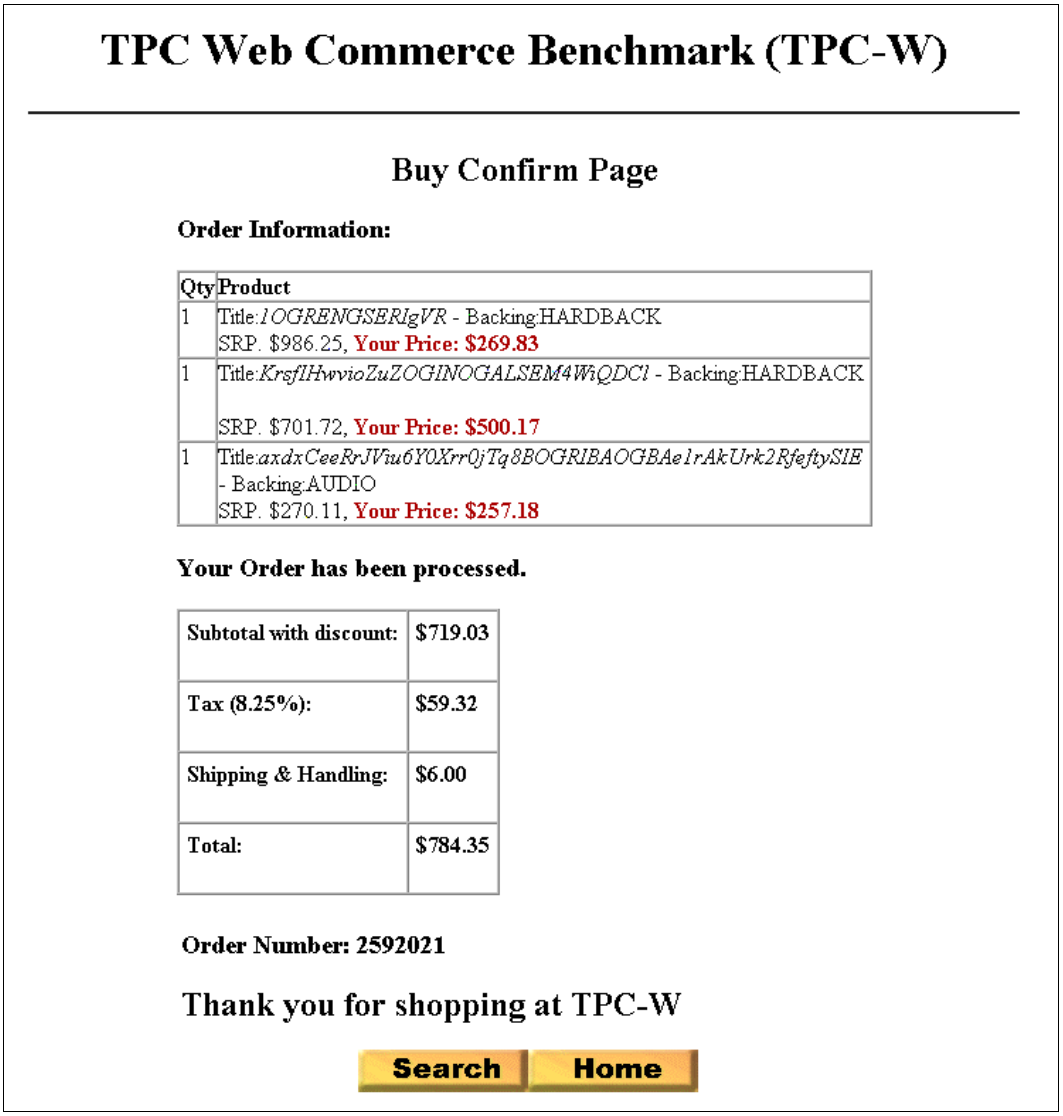
Comment: The application program may implement the above set of atomic operations by including all of the execution steps within a single database transaction.

2.7.3.4 The SUT produces the HTML code for the Buy Confirm page, including the complete order information and the customer information; and returns the page to the EB.

2.7.3.5 The EB selects a navigation option according to Clause 5.2.2.

2.7.4 Response Page Definition

A sample of the Buy Confirm Page is shown below as it would appear in a Browser:



TPC Web Commerce Benchmark (TPC-W)

Buy Confirm Page

Order Information:

Qty	Product
1	Title:*1OGRENGSERIgVR* - Backing:HARDBACK SRP. $986.25, **Your Price: $269.83**
1	Title:*KrsflHwvioZuZOGINOGALSEM4WiQDCl* - Backing:HARDBACK SRP. $701.72, **Your Price: $500.17**
1	Title:*axdxCeeRrJViu6Y0Xrr0jTq8BOGRIBAOGBAe1rAkUrk2RfeftySIE* - Backing:AUDIO SRP. $270.11, **Your Price: $257.18**

Your Order has been processed.

Subtotal with discount:	**$719.03**
Tax (8.25%):	**$59.32**
Shipping & Handling:	**$6.00**
Total:	**$784.35**

Order Number: 2592021

Thank you for shopping at TPC-W

Search Home

The HTML code for the Buy Confirm Page (an example of which can be found in Appendix F) must meet the following requirements:

1. The length of the code must be a minimum of 1,850 characters (excluding spaces).
2. The following web objects must be referenced:
   - Images for the <Home> and <Search> buttons

2.7.5 EB Navigation Options

2.7.5.1 <Home> button: Make an HTTP request to initiate the Home web interaction.

2.7.5.2 < Search> button: Make an HTTP request to initiate the Search Request web interaction.

2.8 Order Inquiry Web Interaction

2.8.1 Overview

This secure web interaction returns to the EB a web page which allows a User to provide the information necessary to enter or confirm their identity as a returning customer. This is the first step in displaying information about the customer's last order.

2.8.2 Input Requirements

2.8.2.1 This web interaction is invoked by an HTTP request and requires the following input data:

- The URL for the Order Inquiry Page, which is contained in the HTTP request.

2.8.3 Processing Definition

2.8.3.1 If this User is a Customer and its C_ID is known, the SUT obtains the following information about the Customer:

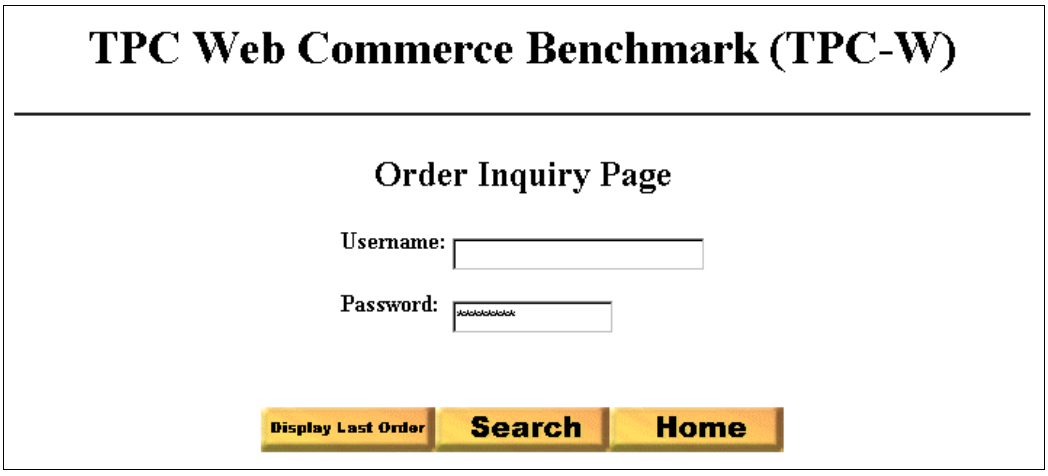
- C_UNAME

2.8.3.2 The SUT produces the HTML code for the Order Inquiry Page, containing C_UNAME if available, and returns the web page to the EB.

2.8.3.3 The EB selects a navigation option according to Clause 5.2.2.

2.8.4 Response Page Definition

A sample of the Order Inquiry Page is shown below as it would appear in a Browser:



The HTML code for the Order Inquiry Page (an example of which can be found in Appendix F) must meet the following requirements:

1. The length of the code must be a minimum of 800 characters (excluding spaces).
2. The following web objects must be referenced:
    - Images for the <Display Last Order>, <Home> and <Search> buttons

2.8.5 EB Navigation Options

2.8.5.1 <Display Last Order> button: Make an HTTP request to initiate the Order Display web interaction, with the following input:

- If a C_ID is known, then:
    - PASSWD corresponding to the Customer currently emulated by the EB
- If a C_ID is not known, then:
    - C_ID is generated by the EB using the method defined in 2.3.2.
    - UNAME is generated by the EB according to Clause 4.6.2.9.
    - PASSWD is generated by the EB according to Clause 4.6.2.10.

2.8.5.2 <Home> button: Make an HTTP request to initiate the Home web interaction.

2.8.5.3 < Search> button: Make an HTTP request to initiate the Search Request web interaction.

2.9 Order Display Web Interaction

2.9.1 Overview

This secure web interaction returns to the EB a web page which displays the status of the last order placed by the customer.

2.9.2 Input Requirements

This web interaction is invoked by an HTTP request and requires the following input data:

- The URL for the Order Display Page, which is contained in the HTTP request.
- The following customer identification information:
    - UNAME
    - PASSWD

2.9.3 Processing Definition

2.9.3.1 The SUT obtains the C_PASSWD from the customer with (C_UNAME = UNAME) and compares PASSWD with C_PASSWD.

Comment: In a real-life application, an invalid password check would result in rejecting the order inquiry. In the context of the benchmark, all supplied passwords will be valid, allowing the processing of the order to continue. The application program is not required to include any logic to deal with invalid passwords.

2.9.3.2 The SUT obtains the following information about the last order of the customer based on its C_UNAME and within a single database transaction:

- O_ID
- C_FNAME
- C_LNAME
- O_DATE
- O_SUB_TOTAL
- O_TAX
- O_TOTAL
- O_SHIP_TYPE
- O_SHIP_DATE
- For the billing address
  - ADDR_STREET1
  - ADDR_STREET2
  - ADDR_STATE
  - ADDR_ZIP
  - ADDR_CO_ID
- For the shipping address
  - ADDR_STREET1
  - ADDR_STREET2
  - ADDR_STATE
  - ADDR_ZIP
  - ADDR_CO_ID
- O_STATUS
- For each item on the order:
  - OL_I_ID
  - I_TITLE
  - I_PUBLISHER
  - I_COST
  - OL_QTY
  - OL_DISCOUNT
  - OL_COMMENT
- For the credit card transaction of the order:
  - CX_TYPE

2.9.3.3 The SUT produces the HTML code for the Order Display Page containing the information obtained and returns the web page to the EB.

Comment: If there are no orders for this Customer, then the Order Display Page is displayed with no order data in it.

2.9.3.4 The EB selects a navigation option according to Clause 5.2.2.

2.9.4 Response Page Definition

A sample of the Order Display Page is shown below as it would appear in a Browser:

TPC Web Commerce Benchmark (TPC-W)

Order Display Page

Order ID: 2592013
Order Placed on 4/1/1999
Shipping Type: Regular
Ship Date: 808464674/141402
Order Subtotal: $968.97
Order Tax: $79.94
Order Total: $1054.91

Bill To:
John Doe

1 Some Place
Appt 1
Dullsville
CA 92691
Libya

Email: me@work.com
Phone: 123-456-7890

Ship To:

1 Some Place
Appt 1
Dullsville
CA 92691
Libya

Credit Card Type: Visa
Order Status: Pending

Item #	Title	Cost	Qty	Discount	Comment
5314	**Title: Xjg1CdU73Q7KQRZ0fd7BASERIOGREmY1JHeT11op0z9m** **Publisher: 9XVWJDR8WGOEps4**	$930.23	1	1.00	comment
62440	**Title: ojPavZt5Co6LW57SEKy7BB22ODy8OGSEALBABA5isiu** **Publisher: 3dGH107y2oxeGOqWxYMajVTk8zw8hZdH7**	$658.25	1	1.00	comment

Search Home

The HTML code for the Order Display Page (an example of which can be found in Appendix F) must meet the following requirements:

1. The length of the code must be a minimum of 1,850 characters (excluding spaces).
2. The following web objects must be referenced:
   - Images for the <Home> and <Search> buttons

2.9.5 EB Navigation Options

2.9.5.1 <Home> button: Make an HTTP request to initiate the Home web interaction.

2.9.5.2 < Search> button: Make an HTTP request to initiate the Search Request web interaction.

2.10 Search Request Web Interaction

2.10.1 Overview

This unprotected web interaction returns to the EB a web page which allows a User to specify search criteria to find qualifying items.

2.10.2 Input Requirements

This web interaction is invoked by an HTTP request and requires the following input data:

- The URL for the Search Request Page, which is contained in the HTTP request.

2.10.3 Processing Definition

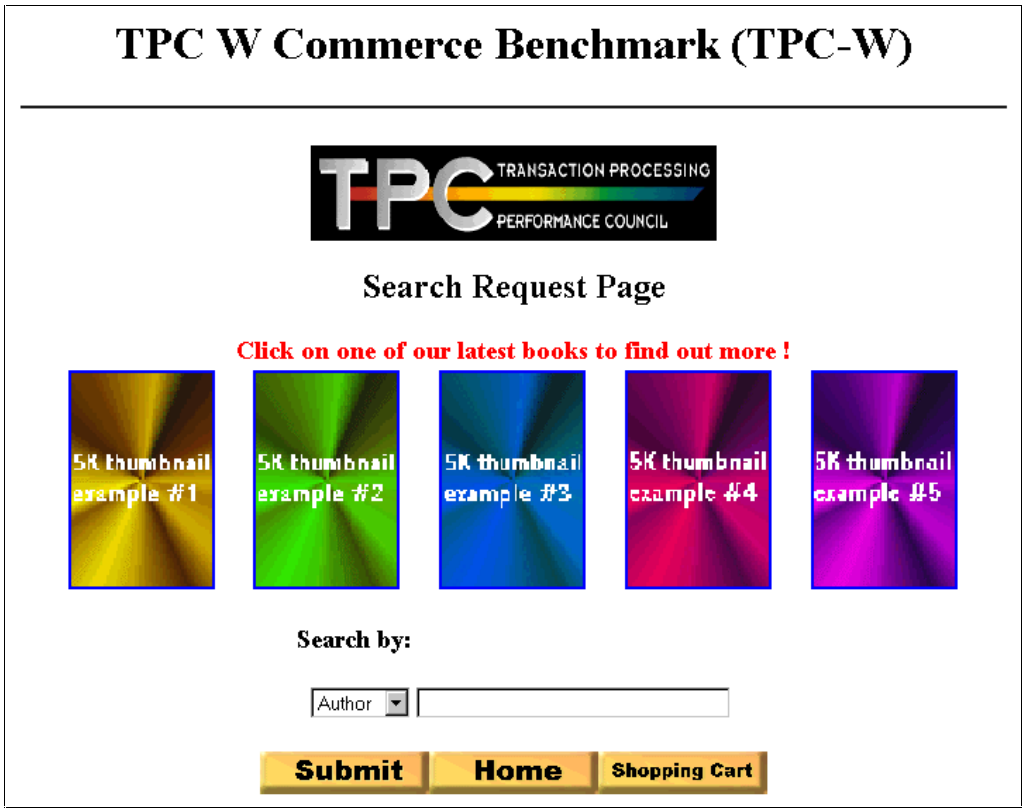
2.10.3.1 The SUT executes the Promotional Processing, as defined in Clause 2.2.17.

2.10.3.2 The SUT produces the HTML code for the Search Request Page and returns the web page to the EB.

2.10.3.3 The EB selects a navigation option according to Clause 5.2.2.

2.10.4 Response Page Definition

A sample of the Search Request Page is shown below as it would appear in a Browser:



The HTML code for the Search Request Page (an example of which can be found in Appendix F) must meet the following requirements:

1. The length of the code must be a minimum of 1,900 characters (excluding spaces).
2. The following web objects must be referenced:
   - Images for the <Shopping Cart>, <Submit> and <Home> buttons
   - Images for the TPC Logo
   - Images for the thumbnail of the five promotional items

2.10.5 EB Navigation Options

2.10.5.1 <Submit> button: Make an HTTP request to initiate the Search Result web interaction, with the following input:

- The EB selects a SEARCH_TYPE from a uniform distribution over the following values: [AUTHOR, TITLE, SUBJECT]
  - If SEARCH_TYPE is selected to be AUTHOR then SEARCH_STRING is set to DigSyl(NURand(A, 0, NUM_ITEMS / 10), 7).
  - **Comment**: SEARCH_STRING is meant to match a substring embedded in the A_LNAME field of the AUTHOR table as described in clause 4.6.2.18. This should yield an average of 2.5 Authors and 10 books.
  - If SEARCH_TYPE is selected to be TITLE then SEARCH_STRING is set to DigSyl(NURand(A, 0, NUM_ITEMS / 5), 7).

- **Comment**: SEARCH_STRING is meant to match a substring embedded in the I_TITLE field of the ITEM table as described in clause 4.6.2.17. This should yield an average of 5 books.
- If SEARCH_TYPE is selected to be SUBJECT then SEARCH_STRING is generated from a uniform random distribution according to Clause 4.6.2.11.

In the above NURand functions, the value for A must be selected from the following table depending on NUM_ITEMS, the number of row in the ITEM table (see clauses 4.3 and 4.7):

For NUM_ITEMS	Value for A
1,000	63
10,000	511
100,000	4,095
1,000,000	32,767
10,000,000	524,287

2.10.5.2 <Home> button: Make an HTTP request to initiate the Home web interaction.

2.10.5.3 <Shopping Cart> button: Make an HTTP request to initiate the Shopping Cart web interaction, with the following input:

- An empty list of (I_ID, QTY) pairs
- The ADD_FLAG is set to "N"

2.11 Search Result Web Interaction

2.11.1 Overview

This unprotected web interaction returns to the EB a web page which contains the list of items that match a given search criteria.

2.11.2 Input Requirements

This web interaction is invoked by an HTTP request and requires the following input data:

- The URL for the Search Result Page, which is contained in the HTTP request.
- The following two fields:
  - SEARCH_TYPE
  - SEARCH_STRING

2.11.3 Processing Definition

2.11.3.1 Of the entire set of (I_ID, I_TITLE) pairs for items that match the selection criteria, sorted by ascending I_TITLE, the first 50 pairs (or less if the entire set contains less than 50 pairs) are obtained. The following search predicates are used (where <string> is the content of SEARCH_STRING):

- If SEARCH_TYPE = “AUTHOR”, the search predicate is:

I_AUTHOR = “%<string>%”

- If SEARCH_TYPE = “TITLE”, the search predicate is:

I_ TITLE = “%<string>%”

- If SEARCH_TYPE = “SUBJECT”, the search predicate is:

I_ SUBJECT = “<string>”

2.11.3.2 The SUT executes the Promotional Processing, as defined in Clause 2.2.17.

2.11.3.3 The SUT produces the HTML code for the Search Result Page containing the fields I_TITLE, A_FNAME, A_LNAME for the list of items obtained with links to the Product Detail pages for these items and returns the web page to the EB.

2.11.3.4 The EB selects a navigation option according to Clause 5.2.2.

2.11.4 Response Page Definition

A sample of the Search Result Page is shown below as it would appear in a Browser:

The HTML code for the Search Result Page (an example of which can be found in Appendix F) must meet the following requirements:

1. The length of the code must be a minimum of 10,050 characters (excluding spaces).
2. The following web objects must be referenced:
    - Images for the <Shopping Cart>, <Search> and <Home> buttons
    - Images for the TPC Logo
    - Images for the thumbnail of the five promotional items

2.11.5 EB Navigation Options

2.11.5.1 <Shopping Cart> button: Make an HTTP request to initiate the Shopping Cart Update web interaction, with the following input:

- An empty list of (I_ID, QTY) pairs
- The ADD_FLAG is set to "N"

2.11.5.2 < Search> button: Make an HTTP request to initiate the Search Request web interaction.

2.11.5.3 <Item Title> link: Make an HTTP request to initiate the Product Detail web interaction, with the following input:

- I_ID, where the EB selects from a uniform random distribution one entry from the list of URLs for obtained items.

2.11.5.4 <Home> button: Make an HTTP request to initiate the Home web interaction.

2.12 New Products Web Interaction

2.12.1 Overview

This unprotected web interaction returns to the EB a web page which contains the list of recently released items.

2.12.2 Input Requirements

This web interaction is invoked by an HTTP request and requires the following input data:

- The URL for the New Products Page, which is contained in the HTTP request.
- The following field:
    - SUBJECT_STRING

2.12.3 Processing Definition

2.12.3.1 Of the entire set of (I_ID, I_TITLE) pairs for items on the selected subject, sorted by descending I_PUB_DATE and ascending I_TITLE, the first 50 pairs (or less if the entire set contains less than 50 pairs) are obtained. The following search predicate is used (where <string> is the content of SUBJECT_STRING):

I_ SUBJECT = "<string>"

2.12.3.2 The SUT executes the Promotional Processing, as defined in Clause 2.2.17.

2.12.3.3 The SUT produces the HTML code for the New Products Page containing the fields I_TITLE, A_FNAME, A_LNAME for the list of items obtained with links to the Product Detail pages for these items and returns the web page to the EB.

2.12.3.4 The EB selects a navigation option according to Clause 5.2.2.

2.12.4 Response Page Definition

A sample of the New Products Page is shown below as it would appear in a Browser:

TPC Web Commerce Benchmark (TPC-W)

New Products Page - Subject: ARTS

Click on one of our latest books to find out more !

	Author	Title
1.	*stU6Plb4ZT6Grh ViBARESEALNGMF6D9On1*	1OGRENGSERIgVR
2.	*EOnhFBqEk1O BANGSEOGBA*	KrsfIHwvioZuZOGINOGALSEM4WiQDCl
3.	*ZnPyLyHaMt axk4BARIOGBABAMBJMo3*	WgAbL4kp5i2RObmj1WtqXKq8OGULULREBAComTAXgHnNqenOpezH
4.	*dRUZWCGd tnpBAOGULOGRIUE*	axdxCeeRrJViu6Y0Xrr0jTq8BOGRIBAOGBAe1rAkUrk2RfeftySIE
5.	*zv7n nBAREATINULBxtrfQ*	1I2uPOGINSEOGNGHQw3gXZvcEAO07UmZzjAbRcCprGJkeBLX
6.	*Jcln seCBARIRIRIRESxF*	MREG0dXuqTnBAREATALOGySi7Nn
7.	*kWYXA BAINRIRERI*	GqvBASENGBAATF7xED7GTevCYGhT
8.	*dGFbrl9vKsoJgKkL CBABAALNGBA0WkKll*	GnYuA8tcUOGALULNGNGZo

Lines deleted for formatting purposes

29.	*w4IZupli7o98rTQ2 gqKA2RFIBAOGSEBABAFU*	N7AOGOGRISESECfBX9rgLFAYzHUO1O4W5FVV1WLFw8K
30.	*dzhXQntc7WaVC BASEULULIN*	d9JSJBCqYYNnBtr3x26BBAATRIRIATJIpQzm4CslKw44z13PeC
31.	*elS0PutEc340RR1FK ZrBASEULSEUL*	SaFjn8mjLjBvYOGSEREALSEEDk4p1dU
32.	*hFd0qwdo BAULBAINAT*	eWD5cfJHZh2HDneIOGALREBAINW0Xnn2G4CKWJ6n2JS6fwsNc
33.	*USa0zHZ2qEzAshTM Jb3kDBAINBAATULL*	2SSUi8BQznlzVb9mtQc2kfzIy9hE3lW5WBARIRERERIYHj13Htfct
34.	*eyubq6Ecs7 ie1BAOGNGRIREOXLNICC*	ykI0C5pmAorOGSEATINNGVanCzJQgHpHI5xeuPXF8wBXuV1YKS
35.	*laBhg4w Z7BARIINRERIlm4aeF6*	6OGOGBAATUL0MOVhMW5k0Z38eDNMuDHywZvKG7Mzhalx6xiZM4SPB9
36.	*0T1hbcP BAALSEULOGcKN*	FPTg2a9q2x1RKhrlQZoGOOGBAREINBApG
37.	*x23ZPje2ANQkpiTQ yjRh17EBABAALBABA*	BR2kTOU3iZ1SI32r9Na4J2r9essqZsHSOGNGBABANGYslmkB5LoIAnqByQZP
38.	*hkPT MYo6RNBAREOGATOGWU*	wG5QOiK7PGXQPkBABANGRISE2F
39.	*gpC8UE BAALNGBAATo*	8Rr5OGINRISEOGgyj
40.	*uOjUkg311fFVHNY c1ymBAINULBABA3H5*	i8pR3xCOGSERIBAULu7
41.	*YKJ7EjFTepimtcKAQuD fEceBARIOGINNGJr6*	CHPHSMbAHOGBARIATAL3An
42.	*WydV6nf wZQBAALSEATBAI3*	BAATALULSE5VvYkV
43.	*AiVg5L5xoIYns4duE lHjkhMGWBABAULNGINLC*	tJ915cy3XXiw5GTiH8KexkdxACCoB6kGepMuupnOGATULRISEX7u
44.	*bD8fP6n1t 6AQcBAULRIRIAT6T*	y47S6pbnFjNcnOGNGREINULomIqoy
45.	*XRtBAGg1Ppai41Z95L jBAALSEULATU*	Pozk6UDuOXUXBAINBAULATX5jciYRwEjLra6ciLVvzZUn7yP
46.	*W8S22 BARESEATRI*	eXtFR1gZnOGINBARISEbHOByUn
47.	*9aoVyxHNyQ 4mBAREOGSEIN6W*	zsit9eiWOGSENGATATTwDHvBM72JaQ0DZAbhomd2Mk6vEWi
48.	*W3T3e7zGBr1mUQw 8mBARIBARIRIxERPcW*	tpf7WVEROGNGALREOG9jpuwr79LYceHk7KhUxjI2VR
49.	*HqVmdTJ9YBr d7N3h4vBAINALRIALQs*	ItMOGOGULOGULuCDFQceP4e20
50.	*1O0t6h ij9d6FFBAREREBAALZi*	E34SctIMqrFdID1L4SBAREULALRIcbHOaAoMvUB

Shopping Cart | Search | Home

The HTML code for the New Products Page (an example of which can be found in Appendix F) must meet the following requirements:

1. The length of the code must be a minimum of 10,450 characters (excluding spaces).
2. The following web objects must be referenced:
    - Images for the <Shopping Cart>, <Search> and <Home> buttons
    - Images for the TPC Logo
    - Images for the thumbnail of the five promotional items

2.12.5 EB Navigation Options

2.12.5.1 <Shopping Cart> button: Make an HTTP request to initiate the Shopping Cart Update web interaction, with the following input:

- An empty list of (I_ID, QTY) pairs
- The ADD_FLAG is set to "N"

2.12.5.2 < Search> button: Make an HTTP request to initiate the Search Request web interaction.

2.12.5.3 <Item Title> link: Make an HTTP request to initiate the Product Detail web interaction, with the following input:

- I_ID, where the EB selects from a uniform random distribution one entry from the list of URLs for obtained items.

2.12.5.4 <Home> button: Make an HTTP request to initiate the Home web interaction.

2.13 Best Sellers Web Interaction

2.13.1 Overview

This unprotected web interaction returns to the EB a web page which contains the list of best seller items.

2.13.2 Input Requirements

This web interaction is invoked by an HTTP request and requires the following input data:

- The URL for the Best Sellers Page, which is contained in the HTTP request.
- The following field:
    - SUBJECT_STRING

2.13.3 Processing Definition

2.13.3.1 Of the entire set of (I_ID, I_TITLE) pairs for items on the selected subject, sorted by ascending I_TITLE, the first 50 pairs (or less if the entire set contains less than 50 pairs) are obtained.. The following search predicate is used (where <string> is the content of SUBJECT_STRING):

I_ SUBJECT = "<string>" for the order lines of the 33,333 most recent orders (where O_ID = OL_O_ID) based on O_DATE and sorted by descending sum(OL_QTY) grouped on OL_I_ID

2.13.3.2 The SUT executes the Promotional Processing, as defined in Clause 2.2.17.

2.13.3.3 The SUT produces the HTML code for the Best Sellers Page containing the fields I_TITLE, A_FNAME, A_LNAME for the list of items obtained with links to the Product Detail pages for these items and returns the web page to the EB.

2.13.3.4 The EB selects a navigation option according to Clause 5.2.2.

2.13.4 Response Page Definition

A sample of the Best Sellers Page is shown below as it would appear in a Browser:

TPC Web Commerce Benchmark (TPC-W)

Best Sellers Page - Subject:ARTS

Click on one of our latest books to find out more !

	Author	Title
1.	*Okvj6t5Frs4ykeK6HJm KO1cfBAOGALULREUIA*	RBOGSESEINNG4NZUJf4YWwV9HK
2.	*iBdVTlii BANGRIREALrpDvElsxL5*	aOowPD30kheFeLgTWflRLlPdcgAOGRESERINGcO
3.	*HgdtVymg UqBASESEULSE35jl*	FqXJOGATINBAULS7r
4.	*wJ5 4OD4BANGRIATOG*	0reJlF8aknzCTuXl6ebuClRgneLngXzTpOdNboOYBAATNGREOG3LmkNXa8
5.	*Qdh4YU6CxQBqFN IMkbVBAOGRIRIRI0IU7*	BABASESENGbH161d71eBbQjm2ZZwEc4yChCDUqN2d9CGw6CIN
6.	*g0fNPGG BAATATNGULNHD*	5Kctp04xSPg9BABARIALOGmWJJLuLwsHv
7.	*WkM80ndHi K4BAALININALrTVS*	OGBABASEULmHtM6iO
8.	*gWcX1jHcTS03sExJ0 NiBARIBAULSE*	LplFpUKBZVSRwYOnQOGATSEATATBb0xINNhZTFUwjy

Lines deleted for formatting purposes

29.	*dO4cZZ CYN1BAULREBAINxgPYG*	KkqAAgey9n79yhJuUGOLBARIBANGATEaYK6Brlok8F4FXuJtWQ9pBFl
30.	*2hlfn3Kpo8qI0X BAALALINBAr*	OGSERIRIINpijNgyx
31.	*B5VD2xhT Mc6BAOGBAOGIN47ih*	AV5tEaRQqdZC8V2tfflOGATALALBAOd35hY2UkClu
32.	*Swq4wQq BANGBABAOG*	dRqfMLg4tRLBzvF3GQxexw2BBANGOGRIAL6Ipf4JKPa3EcnsZvKMZrVbvVVs
33.	*su4sf3WNVEiBC HdBBASEINRENGiv*	9kWUyBAOGATOGSEDf5
34.	*vgHF6 YfYpdBARIOGINOGMSg*	63LBAULINREOG0J4iRrjVidDaJwFb
35.	*9j5Dm BAULULOGNGdV0x*	xBAULULNGULLBWHnpPaSrdngap6xLCnEzyCFrbBq
36.	*52dnV8I BANGATATBA3*	dOGATREBABA5fV8oacR
37.	*Rx91OaSUEZaTcanh 5o5U0IPBASEATNGRIN*	F0yicwO6CCyEYrwTrcSL6hsBAOGOGSEINuPAhmNvUWeNiVlE
38.	*nYroyr0IghsKsV5bNfaJ 8tvXIPBABAINALNG52*	WxAOGALRIULREGFcE10FlGz3Y6dnahzWMhSzFPhANC2VPjdTtcCjZaKKdm
39.	*zyOjTuS38b0p BAATOGALNG4*	iHcFaF5sPXQBSzElbkfGdHHoiGdTzOGNGOGULNGdqG0
40.	*2ba8gnqlElkB02AtUT 9LGdC2x7BANGRIREBAa*	dcmmIDczLRna4fgdnEA31fW7Oy2OmYOGSERIBASESCSeMvgEkH
41.	*tjkl1 1u6DBzY9bvBVI BAATINALALN*	YhZf6VKnEm5Mc4ryqUSBM4DwAOGREOGATATseSxxecOJMp8IBznSnPqrd
42.	*FFpfZJ9 u1eo67cEBAREINSEIN*	3FC6BAINATBAREYFUSCsRa0dhuP35SXEYg21hJcAv5v7Domvv
43.	*udCCITo4KLrRBrOM BARINGNGNG*	LIxQRXBLtFSTTQDIUOGRESEATOGAKDXtoHOR
44.	*zc73jR1cZI PPBABAALRESEs*	8eOGATBAREBAxdEsEcWvoPjhTNSdapCgUt7
45.	*Llh7 rMD3BAATOGINOGF*	3BANGATALALIBYaJptT6QUwlCYJbYUxAS4
46.	*jtrc1 ABAREULNGSEwaC*	aW9hIsV88LTyKXoQSCwoyj4b5cQ02MbOGSEREALNGYgwcu1qLue8Zu0P9hVf
47.	*sfnnEqfD dc1ApBAREINSEOGVc9OQ*	kdJfOGATATSEALHehPeIFR
48.	*i3377IYnv U9aBABABARIULfPwy*	efAzCG69aFZiMlS6LziG1N0GoijFBANGBASEINmL8OWA7DSOKYqjnwGIRc
49.	*RCVlEBrCO1db7p4R9 wBAINNGBAALfulB*	nGEtIsPFH13wyZ2rVBAINATBAALZRTtZiJTBviCTr
50.	*OukiB9awtsRj mO5YmBAATOGULATxBc*	vwnFvnQSiBm6aJQFYZylZu7Wo6qrIlKeYYOGOGINULOGG01Mc

Shopping Cart | Search | Home

The HTML code for the Best Sellers Page (an example of which can be found in Appendix F) must meet the following requirements:

1. The length of the code must be a minimum of 10,500 characters (excluding spaces).
2. The following web objects must be referenced:
   - Images for the <Shopping Cart>, <Search> and <Home> buttons
   - Images for the TPC Logo
   - Images for the thumbnail of the five promotional items

2.13.5 EB Navigation Options

2.13.5.1 <Shopping Cart> button: Make an HTTP request to initiate the Shopping Cart Update web interaction, with the following input:

- An empty list of (I_ID, QTY) pairs
- The ADD_FLAG is set to "N"

2.13.5.2 < Search> button: Make an HTTP request to initiate the Search Request web interaction.

2.13.5.3 <Item Title> link: Make an HTTP request to initiate the Product Detail web interaction, with the following input:

- I_ID, where the EB selects from a uniform random distribution one entry from the list of URLs for obtained items.

2.13.5.4 <Home> button: Make an HTTP request to initiate the Home web interaction.

2.14 Product Detail Web Interaction

2.14.1 Overview

This unprotected web interaction returns to the EB a web page which contains detailed information on a selected item.

2.14.2 Input Requirements

2.14.2.1 This web interaction is invoked by an HTTP request and requires the following input data:

- The URL for the Product Detail Page, which is contained in the HTTP request.
- I_ID

2.14.3 Processing Definition

2.14.3.1 The SUT obtains the following information for the selected I_ID within a single database transaction:

- I_TITLE
- A_FNAME
- A_L_NAME

- I_PUB_DATE
- I_PUBLISHER
- I_SUBJECT
- I_DESC
- The graphic object referenced by I_IMAGE
- I_COST
- I_AVAIL
- I_ISBN
- I_PAGE
- I_BACKING
- I_DIMENSIONS

2.14.3.2 The SUT produces the HTML code for the Product Detail Page containing the information obtained and returns the web page to the EB.

2.14.3.3 The EB selects a navigation option according to Clause 5.2.2.

2.14.4 Response Page Definition

A sample of the Product Detail Page is shown below as it would appear in a Browser:

The HTML code for the Product Detail Page (an example of which can be found in Appendix F) must meet the following requirements:

1. The length of the code must be a minimum of 1,650 characters (excluding spaces).
2. The following web objects must be referenced:
   - Images for the <Add to Cart>, <Search>, <Update> and <Home> buttons
   - Images for the TPC Logo
   - Images for the thumbnail of the five promotional items

2.14.5 EB Navigation Options

2.14.5.1 <Add to Cart> button: Make an HTTP request to initiate the Shopping Cart Update web interaction, with the following input:

- A single (I_ID, QTY) pair, where I_ID is the selected item and QTY is set to 1
- The ADD_FLAG is set to "Y"

2.14.5.2 <Home> button: Make an HTTP request to initiate the Home web interaction.

2.14.5.3 < Search> button: Make an HTTP request to initiate the Search Request web interaction.

2.14.5.4 <Item Title> link: Make an HTTP request to initiate the Product Detail web interaction, with the following input:

- I_ID, where the EB selects from a uniform random distribution one entry from the list of URLs for obtained items.

Note: This option portrays the navigation outcome from a User using the Browser's back-navigation features to go back to the previous page (either the Search Result Page, the New Products Page or the Best Sellers Page) and selecting to see the details of a product. The list of URLs from the previous page is maintained in the CURL of the EB.

2.14.5.5 <Update> button: Make an HTTP request to initiate the Admin Request web interaction with the following input field:

- I_ID

2.15 Admin Request Web Interaction

2.15.1 Overview

This unprotected web interaction returns to the EB a web page which allows a User to request the update of an item.

2.15.2 Input Requirements

This web interaction is invoked by an HTTP request and requires the following input data:

- The URL for the Admin Request Page, which is contained in the HTTP request.
- I_ID

2.15.3 Processing Definition

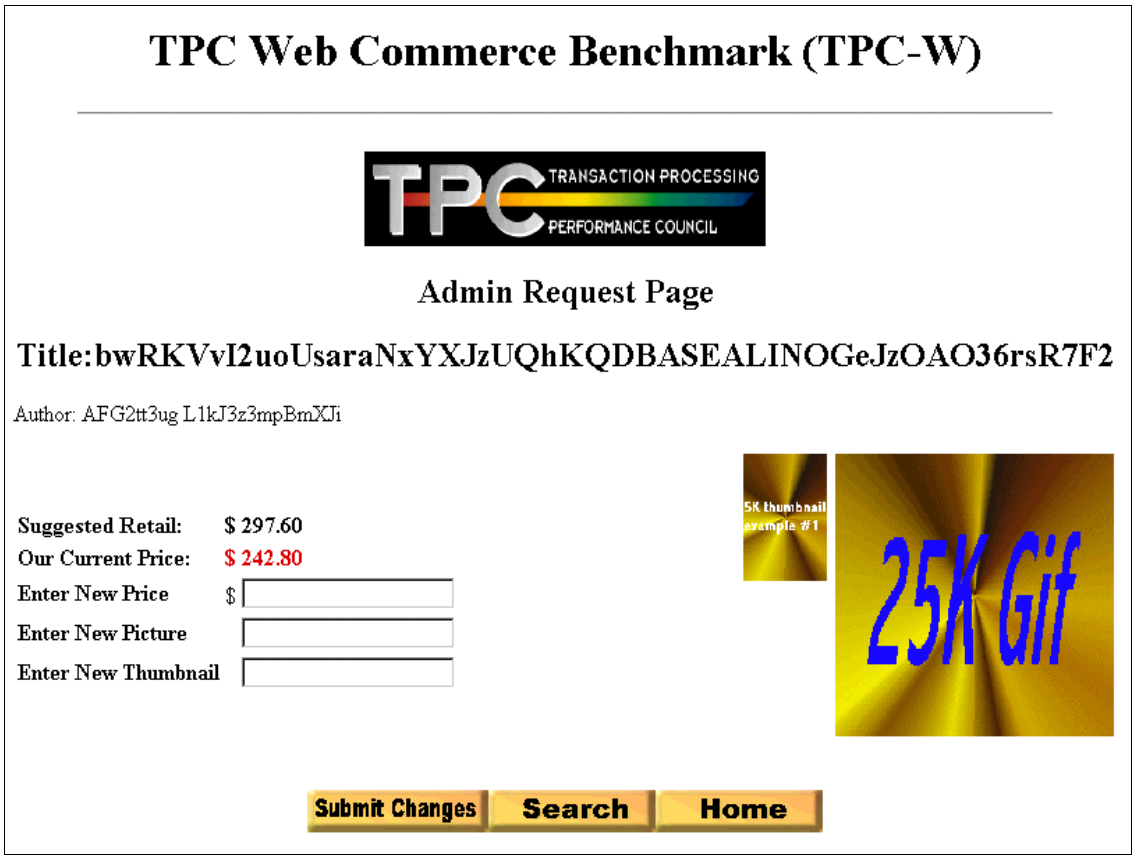
2.15.3.1 The SUT obtains I_SRP and I_COST for the targeted item.

2.15.3.2 The SUT produces the HTML code for the Admin Request Page and returns the web page to the EB.

2.15.3.3 The EB selects a navigation option according to Clause 5.2.2.

2.15.4 Response Page Definition

A sample of the Admin Request Page is shown below as it would appear in a Browser:



The HTML code for the Admin Request Page (an example of which can be found in Appendix F) must meet the following requirements:

1. The length of the code must be a minimum of 1,450 characters (excluding spaces).
2. The following web objects must be referenced:
   - Images for the <Submit Changes>, <Search> and <Home> buttons
   - Images for the TPC Logo
   - Images for the item and its thumbnail

2.15.5 EB Navigation Options

2.15.5.1 <Submit Changes> button: Make an HTTP request to initiate the Admin Confirm web interaction, with the following input:

- The EB generates a values for:
  - I_NEW_IMAGE, generated according to Clause 4.6.2.12 from a random distribution over the range of I_IMAGE.
  - I_NEW_THUMBNAIL, generated according to Clause 4.6.2.12 from a random distribution over the range of I_THUMBNAIL.
  - I_NEW_COST, generated according to Clause 4.7.1 from a random distribution over the range of I_SRP.

2.15.5.2 <Home> button: Make an HTTP request to initiate the Home web interaction.

2.15.5.3 < Search> button: Make an HTTP request to initiate the Search Request web interaction.

2.16 Admin Confirm Web Interaction

2.16.1 Overview

This unprotected web interaction updates an item and returns to the EB a web page which contains the details of the updated item.

2.16.2 Input Requirements

This web interaction is invoked by an HTTP request and requires the following input data:

- The URL for the Admin Confirm Page, which is contained in the HTTP request.
- I_ID
- I_NEW_IMAGE
- I_NEW_THUMBNAIL
- I_NEW_COST

2.16.3 Processing Definition

2.16.3.1 The SUT obtains the following information for the targeted item:

- I_ID
- I_TITLE
- A_FNAME
- A_LNAME
- I_SUBJECT
- I_DESC
- I_COST
- I_IMAGE

2.16.3.2 The SUT updates the targeted item with (I_COST = I_NEW_COST), (I_IMAGE = I_NEW_IMAGE) , (I_ THUMBNAIL = I_NEW_ THUMBNAIL) and (I_PUB_DATE = <current-date>); within a single database transaction.

2.16.3.3 The SUT performs the following processing steps:

- Of all the orders sorted by descending O_DATE, obtain the set of the first 10,000.
- From all orders that include the targeted item in the above set, obtain the set of unique customer that placed these orders.

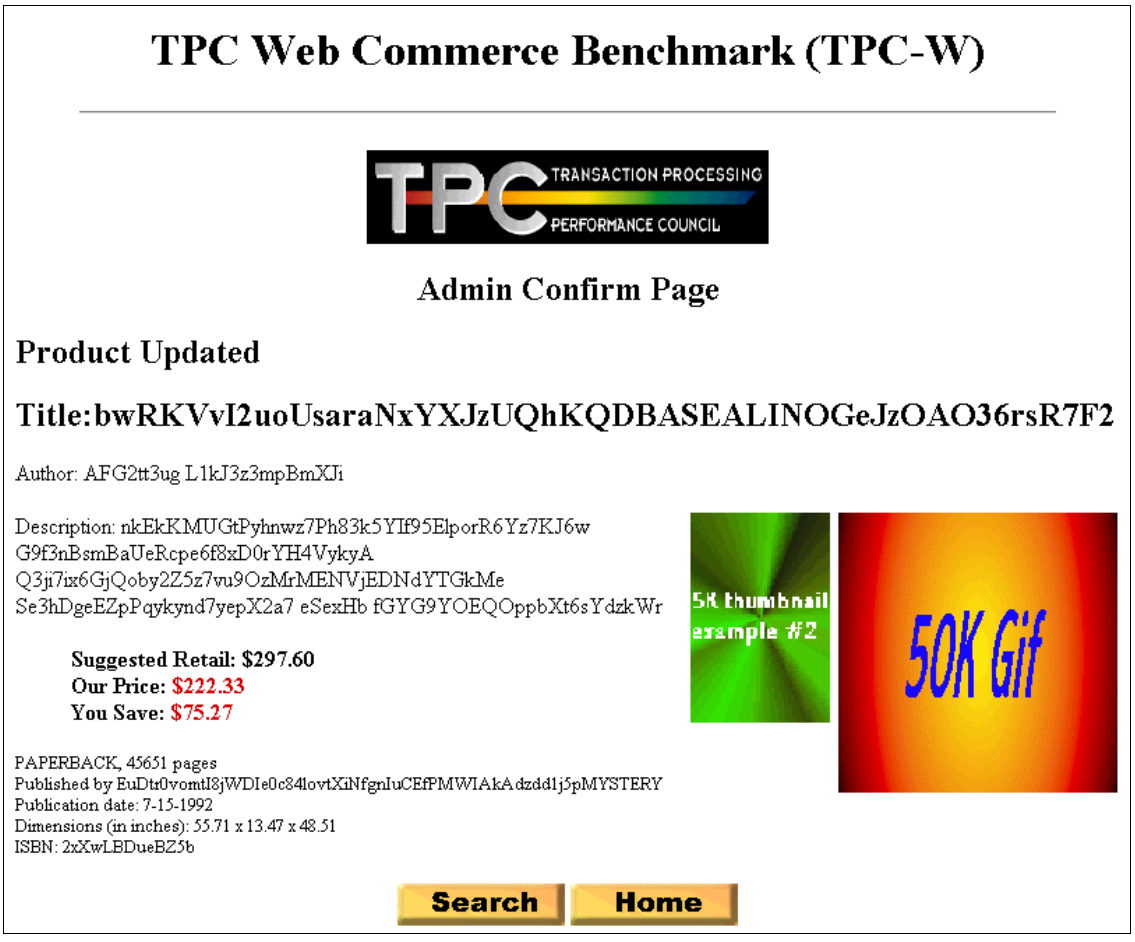
- From all customers in the above set, obtain the list of unique items ordered by these customers within the above set of 10,000 orders, and sort these items by descending aggregated quantity (i.e., sum(OL_QTY) for each unique OL_I_ID).
- From the sorted list of items above, obtain the set (I_ID1, I_ID2, I_ID3, I_ID4, I_ID5) of the first five items.
- Update the targeted item with (I_RELATED1 = I_ID1), (I_RELATED2 = I_ID2), (I_RELATED3 = I_ID3), (I_RELATED4 = I_ID4) and (I_RELATED5 = I_ID5).

2.16.3.4 The SUT produces the HTML code for the Admin Confirm Page and returns the web page to the EB.

2.16.3.5 The EB selects a navigation option according to Clause 5.2.2.

2.16.4 Response Page Definition

A sample of the Admin Confirm Page is shown below as it would appear in a Browser:



The HTML code for the Admin Confirm Page (an example of which can be found in Appendix F) must meet the following requirements:

1. The length of the code must be a minimum of 1,950 characters (excluding spaces).
2. The following web objects must be referenced:

- Images for the <Add to Cart>, <Search> and <Home> buttons
- Images for the TPC Logo
- Images for the item and its thumbnail

2.16.5 EB Navigation Options

2.16.5.1 < Search> button: Make an HTTP request to initiate the Search Request web interaction.

2.16.5.2 <Home> button: Make an HTTP request to initiate the Home web interaction.

2.16.5.3 < Search> button: Make an HTTP request to initiate the Search Request web interaction.

Clause 3 - DATABASE TRANSACTION AND SYSTEM PROPERTIES

3.1 Database Transaction ACID Properties

It is the intent of this section to define the ACID properties requirements for database transactions and to specify a series of tests that must be performed to demonstrate that these requirements are met.

3.1.1 Introduction

3.1.1.1 All interactions with any database maintaining the tables defined in Clause 1 must be made through a database transaction supporting full ACID properties, as defined in Clauses 3.1.2 to 3.1.4..

3.1.1.2 No finite series of tests can prove that the ACID properties are fully supported. Passing the specified tests is a necessary, but not sufficient, condition for meeting the ACID requirements. However, for fairness of reporting, only the tests specified here are required and must appear in the Full Disclosure Report for this benchmark.

Comment: These tests are intended to demonstrate that the ACID properties are supported by the SUT and enabled during the performance Measurement Interval. They are not intended to be an exhaustive quality assurance test.

3.1.1.3 All mechanisms needed to insure full ACID properties must be enabled during both the ACID test period and the Test Runs (as defined in Clause 5.5). For example, if the system under test relies on undo logs, then logging must be enabled for all database transactions. When this benchmark is implemented on a distributed system, tests must be performed to verify that distributed database transactions (database transactions that are processed on two or more nodes) support the ACID properties.

3.1.1.4 Although the ACID tests may not exercise all types of TPC-W database transactions, the ACID properties must be satisfied for all types.

3.1.1.5 Test sponsors reporting TPC-W results on several systems may perform the ACID tests on any one system for which results are being disclosed, provided that they use the same software executables (e.g., Operating System, Database Manager, Web Server, Application Server, transaction programs). However, the durability tests for system failure (see Clauses 3.1.5.3) must be run on all the systems that are measured. All Full Disclosure Reports must identify the systems used to verify ACID requirements and full details of the ACID tests conducted and results obtained.

Comment: This clause would be applicable, for example, when results are reported for multiple systems in a product line.

3.1.2 Atomicity

3.1.2.1 Atomicity Property Definition

The system under test must guarantee that database transactions are atomic. Within a database transaction the system will either perform all individual operations on the data, or will assure that no operations leave any effects on the data.

3.1.2.2 Atomicity Tests

The atomicity tests require that the Admin Confirm web interaction be instrumented so that the update to the database may be aborted while in progress, affecting the final outcome of the update, but without affecting the ability of the SUT to complete the web interaction.

The following steps describe the atomicity test 1:

1. Request and complete a Best Sellers, a New Products or a Search Result web interaction and choose an item from the response page.
2. Request and complete a Product Detail web interaction for the item chosen in Step 1.
3. Request an instrumented Admin Confirm web interaction for the item chosen in Step 1, changing both the price and the image of the item, and complete the web interaction without aborting its updates.
4. Request and complete a Product Detail web interaction for the item chosen in Step 1.
5. The information presented in the response page from Step 4 must match the update requested in Step 3.

The following steps describe the atomicity test 2:

1. Request and complete a Best Sellers, a New Products or a Search Result web interaction and choose an item from the response page.
2. Request and complete a Product Detail web interaction for the item chosen in Step 1.
3. Request an instrumented Admin Confirm web interaction for the item chosen in Step 1, changing both the price and the image of the item, and abort its updates before completion.
4. Request and complete a Product Detail web interaction for the item chosen in Step 1.
5. The information presented in the response page from Step 4 must match the information presented in Step 2.

3.1.3 Consistency

3.1.3.1 Consistency Property Definition

The system under test must guarantee that database transactions are consistent. Assuming that the database is initially in a consistent state, the system will ensure that any TPC-W database transaction takes the database from one consistent state to another.

3.1.3.2 Consistency Conditions

A consistent state for the TPC-W database is defined to exist when:

1. (I_A_ID) is a valid Foreign Key reference to an existing (A_ID)
2. (C_ADDR_ID) is a valid Foreign Key reference to an existing (ADDR_ID)
3. (O_C_ID) is a valid Foreign Key reference to an existing (C_ID)
4. O_BILL_ADDR and O_SHIP_ADDR are valid Foreign Key references to an existing (ADDR_ID)
5. (OL_I_ID) is a valid Foreign Key reference to an existing (I_ID)
6. (OL_O_ID) is a valid Foreign Key reference to an existing (O_ID)

7. (CX_O_ID) is a valid Foreign Key reference to an existing (O_ID)

8. (CX_CO_ID) is a valid Foreign Key reference to an existing (CO_ID)

9. CX_XACT_AMT = O_TOTAL when CX_O_ID = O_ID

10. (ADDR_CO_ID) is a valid Foreign Key reference to an existing (CO_ID)

3.1.3.3 A TPC-W database, when populated as defined in Clause 4.7, must meet the consistency condition defined in Clause 3.1.3.2.

3.1.3.4 If data is replicated, as permitted under Clause 1.6.6, each copy must meet the consistency condition defined in Clause 3.1.3.2.The implementation of the web interactions must ensure that all consistency conditions defined in Clause 3.1.3.2 are maintained without relying on a limited range of input data. But the implementation of the benchmark is not required to maintain these consistency conditions under arbitrary database transactions.

Comment: This implies that no referential integrity is required to be enforced at the database level.

3.1.3.5 Consistency Test

The verification of the consistency between the CUSTOMER and ADDRESS tables, between the ORDERS and CUSTOMER tables, and between the ORDERS and CC_XACTS tables is done as part of each Durability test (see Clause 3.1.5.5).

While maintaining other consistency conditions defined in Clause 3.1.3.2 is required, their verification is optional.

3.1.4 Isolation

3.1.4.1 Isolation Property Definition

Isolation can be defined in terms of phenomena that can occur during the execution of concurrent database transactions. The following phenomena are considered, given two atomic database transactions, T1 and T2:

P0 ("Dirty Write"): Database transaction T1 reads a data element and modifies it. Database transaction T2 then modifies or deletes that data element, and performs a COMMIT. If T1 were to attempt to re-read the data element, it may receive the modified value from T2 or discover that the data element has been deleted.

P1 ("Dirty Read"): Database transaction T1 modifies a data element. Database transaction T2 then reads that data element before T1 performs a COMMIT. If T1 were to perform a ROLLBACK, T2 will have read a value that was never committed and that may thus be considered to have never existed.

P2 ("Non-repeatable Read"): Database transaction T1 reads a data element. Database transaction T2 then modifies or deletes that data element, and performs a COMMIT. If T1 were to attempt to re-read the data element, it may receive the modified value or discover that the data element has been deleted.

P3 ("Phantom"): Database transaction T1 reads a set of values N that satisfy some <search condition>. Database transaction T2 then executes statements that generate one or more data elements that satisfy the <search condition> used by database transaction T1. If database transaction T1 were to repeat the initial read with the same <search condition>, it obtains a different set of values.

The following table defines four isolation levels with respect to the phenomena P0, P1, P2, and P3.

Isolation Level	P0	P1	P2	P3
0	Not Possible	Possible	Possible	Possible
1	Not Possible	Not Possible	Possible	Possible
2	Not Possible	Not Possible	Not Possible	Possible
3	Not Possible	Not Possible	Not Possible	Not Possible

The following database transactions are defined:

- T_r = Any read-only database transaction used to implement a TPC-W web interaction
- T_u = Any update database transaction used to implement a TPC-W web interaction
- T_n = Any arbitrary transaction (Although arbitrary, this transaction may not do dirty writes)

The system under test will ensure that the isolation requirements defined in the table below are met by all database transactions.

Req. #	For transactions in this set:	these phenomena:	must NOT be seen by this transaction:	Textual Description:
1.	$\{T_u, T_u\}$	P0, P1, P2, P3	T_u	Level 3 isolation between any two TPC-W update transactions.
2.	$\{T_u, T_n\}$	P0, P1, P2	T_u	Level 2 isolation for any TPC-W update transactions relative to any arbitrary transaction.
3.	$\{T_r, T_n\}$	P0, P1	T_n	Level 1 isolation for any TPC-W read-only transaction relative to any arbitrary transaction.

3.1.4.2 Isolation Tests

The isolation tests require that several web interactions be modified so that a query or an update to the database may be halted while in progress, without affecting the final outcome of the query or the update, and without affecting the ability of the SUT to complete the web interaction.

3.1.4.2.1 Isolation Test 1

To verify isolation between two TPC-W update transactions, perform the following steps:

Step 1. From browser A, request and complete a Best Sellers, a New Products or a Search Result web interaction and choose an item from the response page.

Step 2. From browser A, request and complete a Product Detail web interaction for the item chosen in the previous step.

Step 3. From browser B, repeat steps 1 and 2 above for the same item.

Step 4. From browser A, request an Admin Request web interaction from the Product Detail Page of step 2.

Step 5. From browser A, modify the item's image and price and confirm the item's update by requesting an instrumented Admin Confirm web interaction.

Step 6. Interrupt the processing of the Admin Confirm web interaction from browser A after updating the image, but before updating the price.

Step 7. From browser B, repeat step 4 above. Verify that the Admin Request web interaction either waits for browser A to resume, or completes. If the Admin Request web interaction completes, then repeat step 5 above, updating the item's thumbnail and price (using a different value than browser A).

Step 8. Verify that browser B's Admin Confirm web interaction, if requested in step 7, either waits for browser A to resume, or displays an Admin Confirm Page containing the item's image from step 2 and the price and thumbnail as updated by browser B.

Step 9. Resume the processing of the Admin Confirm web interaction interrupted in step 6.

Step 10. Verify that browser A displays an Admin Confirm Page containing the item's new image, and either the old thumbnail and the price as updated by browser A, or the new thumbnail and the price as updated by browser B.

3.1.4.2.2 Isolation Test 2

To verify isolation between a TPC-W update transaction and an arbitrary transaction, perform the following steps:

Step 1. Using a database query and update utility, add a new item to the database with a unique title and commit this update.

Step 2. From a browser, request and complete a Search Result web interaction for the title used in step 1.

Step 3. From the same browser, request and complete a Product Detail web interaction for the item found in the previous step.

Step 4. From the same browser, request an Admin Request web interaction from the Product Detail Page of the previous step.

Step 5. From the same browser, modify the item's image, thumbnail and price and confirm the item's update by requesting an instrumented Admin Confirm web interaction.

Step 6. Interrupt the processing of the Admin Confirm web interaction after updating the image, but before updating the price and the thumbnail.

Step 7. Using a database query and update utility, request the deletion of the item added in step 1. Commit this update as soon as allowed by the database. (The delete request may hang waiting for the processing of the Admin Confirm web interaction to complete.)

Step 8. Resume the processing of the Admin Confirm web interaction interrupted in step 6.

Step 9. Verify that the browser displays an Admin Confirm Page containing the item's new image, new thumbnail and new price.

Step 10. Using a database query and update utility, verify that the item added in step 1 no longer exists in the database.

3.1.4.2.3 Isolation Test 3

To verify isolation between a TPC-W read-only transaction and an arbitrary transaction, perform the following steps:

Step 1. Using a database query and update utility, add a new item to the database with a unique title and commit this update.

Step 2. From a browser, request and complete a Search Result web interaction for the title used in step 1.

Step 3. From the same browser, request and complete a Product Detail web interaction for the item found in the previous step.

Step 4. Using a database query and update utility, request an update of the price of the item added in step 1, but do not commit this update.

Step 5. From the same browser, request and complete another Search Result web interaction for the title used in step 1. If this page is displayed, then request a Product Detail web interaction for the item found here (which should be the same item as found in step 2).

Step 6. Verify that either the browser's Search Result or Product Detail web interaction either waits for the update utility to commit, or displays a Product Detail Page containing the same information as displayed in step 3.

Step 7. Using a database query and update utility, as part of the same database transaction as step 4, request an update of the thumbnail of the item added in step 1, and commit all pending updates to the database.

Step 8. If the browser was waiting in step 6, verify that it has resumed and that it displays either a Search Result page with the same information as displayed in step 2, or a Product Detail Page. If it is a Search Result Page, then request the Product Detail web interaction for the item. The Product Detail Page displayed must contain the new price and thumbnail (as updated in steps 4 and 7) for the item.

Step 9. Using a database query and update utility, request the deletion of the item added in step 1, commit the update to the database and verify that the item no longer exists.

3.1.5 Durability

3.1.5.1 Durability Property Definition

The system under test must guarantee that database transactions are durable. The system will preserve the effects of any committed database transaction after recovery from any single point of failure.

Comment: No system provides complete durability (i.e., durability under all possible types of failures). The specific set of single failures addressed in Clause 3.1.5.3 is deemed sufficiently significant to justify demonstration of durability across such failures.

3.1.5.2 Committed Database Transaction Definition

A database transaction is considered committed when the transaction manager component of the system has either written the log or written the data for the committed updates associated with the database transaction to a durable medium.

Comment: Database transactions can be committed without the user subsequently receiving notification of that fact, as the web interaction may fail after the database transaction has been committed.

3.1.5.3 List of Single Points of Failure

Medium Failure: Permanent irrecoverable failure of any single durable medium containing TPC-W database tables or recovery log data.

If main memory is used as a durable medium, then it must be considered as a potential single point of failure. Sample mechanisms to survive single durable medium failures are database archiving in conjunction with a redo (after image) log, and mirrored durable media. If memory is the durable medium and mirroring is the mechanism used to ensure durability, then the mirrored memories must be independently powered.

Memory Failure: Failure of all or part of memory (loss of contents).

This implies that all or part of memory has failed. This may be caused by a loss of external power or the permanent failure of a board equipped with memory.

System Failure: Instantaneous interruption (system crash/system hang) in processing which requires system re-boot to recover.

This implies abnormal system shutdown which requires loading of a fresh copy of the operating system from the boot device. It does not necessarily imply loss of volatile memory. When the recovery mechanism relies on the pre-failure contents of volatile memory, the means used to avoid the loss of volatile memory (e.g., an Un-interruptible Power Supply) must be included in the system cost calculation. A sample mechanism to survive an instantaneous interruption in processing is an undo/redo log.

3.1.5.4 Durable Medium Definition

A durable medium is a data storage medium that is either:

- An inherently non-volatile medium (e.g., magnetic disk, magnetic tape, optical disk, etc.) or
- A volatile medium that will ensure the transfer of data automatically, before any data is lost, to an inherently non-volatile medium after the failure of external power independently of reapplication of external power. (A configured and priced Un-interruptible Power Supply (UPS) is not considered external power.)

Comment: A durable medium can fail; this is usually protected against by replication on a second durable medium (e.g., mirroring) or logging to another durable medium. Memory can be considered a durable medium if it can preserve data long enough to satisfy the requirement stated in item 2 above, for example, if it is accompanied by an Un-interruptible Power Supply, and the contents of memory can be transferred to an inherently non-volatile medium during the failure. Note that no distinction is made between main memory and memory performing similar permanent or temporary data storage in other parts of the system (e.g., disk controller caches).

3.1.5.5 Durability Tests

For each component susceptible to one of the failure types defined in Clause 3.1.5.3. perform the following steps:

Step 1. Verify that Consistency Conditions 2, 3, 8 and 9, as specified in Clause 3.1.3.2, are met.

Step 2. Obtain the total number of rows in the ORDERS table to determine the current count of orders (count1) in the database.

Step 3. Start the mix of web interactions used for the Shopping Interval (see Clause 5.2.1) from a number of EBs within at least 10% of the number of EBs used for the reported WIPS metric.

Step 4. Run for at least 5 minutes once all EBs have started requesting web interactions; and keep a count of the number of Buy Confirm web interactions successfully completed by all EBs.

Step 5. Cause the failure selected from the list in Clause 3.1.5.3.

Step 6. Stop the RBE and collect the total number of Buy Confirm web interactions successfully completed by all EBs (RBE-count).

Step 7. If necessary, stop and restart the system under test using normal recovery procedures, where applicable.

Step 8. Repeat step 2 to determine the current count of orders (count2) in the database. Verify that (count2-count1) is greater than or equal to the number of successfully completed Buy Confirm web interactions (RBE-count). If there is an inequality, the difference must be less than or equal to the number of EBs active during this test.

Comment: This difference should be due only to database transactions which were committed on the system under test, but for which the Response Page was not returned to the EB before the failure.

Step 9. Verify that Consistency Conditions 2, 3, 8 and 9, as specified in Clause 3.1.3.2, are still met.

3.2 Web Page Consistency Requirements

Most web pages returned by the SUT to the EB reflect the content of the database by displaying plain data and web objects (GIF pictures). As database transactions update the content of the database, the web pages must display a consistent reflection of these updates.

3.2.1 Web Page Consistency Property Definition

3.2.1.1 The effects of any update database transaction must be reflected with consistency in subsequent web pages returned by the SUT to the EB: the web pages must reflect either the effect of the entire update operation or none of it.

3.2.1.2 For the purposes of this definition, a web interaction resulting in an update database transaction can be considered to proceed in three phases:

1. Before the web interaction: the request has not been sent to the SUT. In this phase none of the effects of the update database transaction can be visible in any web page returned by the SUT.

2. During the web interaction: the request has been sent to the SUT, but the response page has not been received by the EB. In this phase, any web pages returned by the SUT must either reflect the entire effects of the update database transaction or must reflect none at all.

3. After the web interaction: the response page has been received in full by the EB. In this phase, the effects of the entire update database transaction must now be reflected in any web page returned by the SUT in response to requests generated after the update.

3.2.2 Web Page Consistency Test

This test requires that a series of web interactions be requested from several User Sessions while the system is operating under the following load:

- Load 1: The mix of web interactions used during the Shopping Interval and producing a WIPS rate greater than both 10% of the reported WIPS and 100 WIPS (or within 90% of the reported WIPS rate if it is less than 100).
- Load 2: A set of 30 different User Sessions, each executing the Product Detail web interaction in a continuous loop. All User Sessions must repeatedly query the same item, chosen in step 1 below, and must maintain a log of the response pages, including the date header provided by the HTTP 1.1 protocol.

The following list of steps must be executed in sequence:

Step 1. From User Session A, complete a New Products web interaction and choose an item from the response page, excluding the first ten items.

Step 2. Start the Load 1 above.

Step 3. Start the load 2 above.

Step 4. Wait a minimum of two minutes.

Step 5. From User Session A, complete an Admin Confirm web interaction for the item chosen in step 1, changing the price, the published date and the image of the item.

Step 6. From User Session B, immediately after step 5 completes, complete a New Products web interaction on the subject of the item chosen in step 1.

Step 7. Wait a minimum of two minutes.

Step 8. Terminate the load 2 and then the load 1.

The following conditions must be met:

- All Product Detail Pages returned by the load 2 meet the requirements defined in Clause 3.2.1.2.
- The New Products Page returned in step 8 contains the item chosen in step 1 within its first ten entries.

Clause 4 - SCALING AND DATABASE POPULATION

4.1 General Scaling Rule

The throughput of the TPC-W benchmark is driven by the activity of the Emulated Browsers (EBs) connected to the store front. Each EB corresponds to a User Session. To increase the throughput demand on the SUT, the number of EBs configured has to be increased. The store front requires a number of rows to populate the tables of the database along with some storage space to maintain static HTML pages, web objects and the data generated during a defined period of activity called the 180-day period. The following requirements define how storage space and database population scale with throughput.

4.1.1 The intent of the scaling requirements is to maintain the ratio between the transaction load presented to the SUT, the cardinality of the tables accessed by the transactions, the required space for storage, and the number of EBs generating the transaction load.

4.1.2 Should any scaling value in Clause 4.3 be exceeded, the others must be increased proportionally to maintain the same ratios among them as in Clause 4.3.

4.1.3 The reported throughput may not exceed the maximum allowed by the scaling requirements in Clause 4.2 and the pacing requirements in Clause 5.2. While the reported throughput may fall short of the maximum allowed by the configured system, the price/performance computation (see Clause 7.1) must report the price for the system as actually configured.

4.2 Scaling Requirements

4.2.1 Database scaling is defined by the cardinality (number of rows) of the ITEM table and the number of EBs, i.e., it is defined by the size of the store and size of the supported customer population.

4.2.2 The cardinality of the ITEM table, NUM_ITEMS, must be chosen from the set of defined scale factors as follows:

1,000; 10,000; 100,000; 1,000,000; 10,000,000

Test sponsors must choose the store front size they want to benchmark by selecting a scale factor from the above defined series.

4.2.3 The cardinality of the AUTHOR table is a function of the ITEM table, as defined in 4.3

4.2.4 The cardinality of the COUNTRY table is fixed.

4.2.5 The cardinality of the other tables is a function of the number of EBs, as defined in 4.3

4.2.6 The reported WIPS throughput is required to satisfy the following inequalities:

(number of EBs)/14 < WIPS < (number of EBs)/7

Comment: The intent of this clause is to prevent reporting a throughput that exceeds the maximum, where the maximum throughput is achieved with infinitely fast web interactions resulting in a null response time and minimum required think times. This is computed to be 1/7 WIPS per EB. To prevent over-scaling the SUT, the throughput cannot fall short of the above 1/14 WIPS per EB, which represents 50% of the computed maximum throughput.

4.3 Configuration

The following scaling requirements represent the initial configuration for the test described in Clause 5:

1. For each table that composes the database, the cardinality of the initial population is specified as follows:

Table Name	Cardinality (in rows)	Typical Row Length (in bytes)	Typical Table Size (in bytes)
CUSTOMER	2880 * (number of EB)	760	2,188,888 k
COUNTRY	92	70	6.44 k
ADDRESS	2 * CUSTOMER	154	887,040 k
ORDERS	.9 * CUSTOMER	220	887,040 k
ORDER_LINE	3 * ORDERS	132	570,240 k
AUTHOR	.25 * ITEM	630	1,575 k
CC_XACTS	1 * ORDERS	80	207,360 k
ITEM	1k, 10k, 100k, 1M, 10M	860	8,600 k

Note 1: Table sizes are computed for 1,000 EBs and 10,000 items
Note 2: Values for the ITEM table do not include the item's Image and Thumbnail
Note 3: The typical row lengths and table sizes given above are examples of what could result from an implementation. They are not requirements. They do not include storage and access overheads.

2. The increment (granularity) for scaling the EB population is one browser.
3. Typical lengths and sizes given here are examples of what could result from an implementation. They are not requirements. They do not include storage/access overheads.
4. The symbol "k" means one thousand and "M" means one million.

4.4 180-Day Space Computation

The storage space required for the database for the 180-day period must be determined as follows:

a) The test database must be built including the initial database population (see Clause 4.4) and all indices present during the test.

b) The test database must be built to sustain the reported throughput during an eight hour period. This excludes performing on the database any operation that does not occur during the Measurement Interval (see Clause 5.5).

c) The growth of the database, G in bytes, should be measured as the initial size of the database compared against the size of the database at the end of theTest Run (see Clause 5.5.1).

d) Assuming TI is the total number of web interactions processed during the duration of the Test Run, and WIPS is the reported throughput, 180-day space in bytes is calculated as follows:

180-Day-Space = Initial Space + ((G / TI) * WIPS * 3600 * 8 * 180)

e) The free space present in the test database is considered as part of the 180-Day-Space.

4.5 Web Server Access Log Requirements

There must be enough space configured on the SUT to store Web Server access logs, in Common Log Format, as specified in Clause 2.2.13, for a period of 8 hours. The space required for this is determined as follows:

a) The growth of the log file, L, is measured as the size of the log file at the beginning of the test duration compared against the size of the log file at the end of the test duration. At the end of the test, it should be verified that the data for the log file is completely written to durable media.

b) Assuming TI is the total number of web interactions processed during the test duration and WIPS is the reported throughput, 8-hour-web-log-space is then computed as follows:

8-hour-web-log-space = (L / TI) * WIPS * 3600 * 8

Comment: In addition to maintaining the Web Server access log for a minimum of 8 hours, the SUT must comply with the requirement for 14 days of uninterrupted execution, as defined in Clause 6.6.1. For example, the requirement implies that the Web Server access log can either be switched and backed-up without system interruption every 8 hours, or that 14 days of Web Server access log must be maintained.

4.6 Database Population

4.6.1 The test described in Clause 5 requires that the database be properly scaled with the initial population. It is allowed, but not required, to reload or rollback the database to its initial population before any Test Run. No other alteration to the defined database population is allowed at any time other than the use of one of the web interaction mixes defined in Clause 5.2.1.

4.6.2 Definition of Terms

4.6.2.1 The term **random** means independently selected and uniformly distributed over the specified range of values.

4.6.2.2 The notation **random a-string [*x* .. *y*]** and **n-string [*x* .. *y*]** represents a string of random alphanumeric and numeric characters respectively, of a random length of minimum *x*, maximum *y*, and mean $(y+x)/2$.

Comment 1: The character set used must include all letters in both upper and lower case, all numeric digits from 0 to 9 inclusive and all of the following special characters including a blank space:

```
! @ # $ % ^ & * ( ) _ - = + { } [ ] | : ; , . ? / ~
```

Comment 2: Generating such strings can be implemented by the concatenation of two strings selected at random from two separate arrays of strings, and where:

1. Both arrays contain a minimum of 10 different strings of characters.
2. The first array contains strings of *x* characters.
3. The second array contains strings of lengths uniformly distributed between zero and (*y* - *x*) characters.

4.6.2.3 The notation unique within [x] represents any one value within a set of x contiguous values, unique within the group of rows being populated. When several groups of rows of the same type are populated, each group must use the same set of x contiguous values.

4.6.2.4 The notation random within [x .. y] represents a random value independently selected and uniformly distributed between x and y, inclusively, with a mean of (x+y)/2, and with the same number of digits of precision as shown. For example, [0.01 .. 100.00] has 10,000 unique values, whereas [1 ..100] has only 100 unique values.

4.6.2.5 The notation random permutation of [x .. y] represents a sequence of numbers from x to y arranged into a random order. This is commonly known as a permutation (or selection) without replacement.

4.6.2.6 The notation random date represents a date value selected at random within the specified range of dates.

4.6.2.7 The notation numeric [x.y digits] represents a floating point number with y digits after the decimal point and x-(y+1) digits before. For example, 5.2 digits implies all values between 0.00 to 99.99.

4.6.2.8 The term DigSyl function is used in this specification to refer to the following function:

DigSyl(D, N) where:

D is a positive integer.

N is an integer that specifies the number of 2-character syllables in the string returned by DigSyl.

N can be null or must be greater than or equal to the number of digits in the decimal representation of D.

DigSyl(D, N) returns a string which is the concatenation of 2-character syllables constructed by replacing each digit in the decimal representation of D with the corresponding 2-character syllable from the following table:

Digit	0	1	2	3	4	5	6	7	8	9
Syllable	BA	OG	AL	RI	RE	SE	AT	UL	IN	NG

If N is greater than the number of digits in the decimal representation of D, then the decimal representation is padded with leading zeros until the total number of decimal digits is N.

Example: To obtain DigSyl(15, 5), 15 is padded to 00015, and then mapped into this concatenation of syllables: BABABAOGSE. (i.e., DigSyl(15, 5) = BABABAOGSE)

If N is 0, then the number of syllables in DigSyl(D, 0) is the number of decimal digits in the representation of D.

Example: To obtain DigSyl(15, 0), 15 is mapped into: OGSE. (i.e., DigSyl(15, 0) = OGSE)

4.6.2.9 The customer user name (**C_UNAME**) must be generated as the string returned by DigSyl(C_ID, 0).

Example: Given a C_ID of 3719, C_UNAME is generated as: DigSyl(3719, 0) = RIULOGNG.

Comment: Because C_IDs are unique numbers, DigSyl associates to each C_ID a unique C_UNAME.

4.6.2.10 The customer password (**C_PASSWD**) must be generated as the string returned by DigSyl(C_ID, 0) converted to all lower case characters.

Example: Given a C_ID of 3719, the resulting C_UNAME is RIULOGNG and the resulting C_PASSWD is `riulogng`.

4.6.2.11 The item subject (**I_SUBJECT**) must be chosen at random from the following list:

ARTS	HOME	RELIGION
BIOGRAPHIES	HUMOR	ROMANCE
BUSINESS	LITERATURE	SELF-HELP
CHILDREN	MYSTERY	SCIENCE-NATURE
COMPUTERS	NON-FICTION	SCIENCE-FICTION
COOKING	PARENTING	SPORTS
HEALTH	POLITICS	YOUTH
HISTORY	REFERENCE	TRAVEL

4.6.2.12 The item image and thumbnail (I_IMAGE, I_THUMBNAIL) are graphic objects. In the event that graphic objects are not stored directly in the database, then the actual location of the objects, either in terms of the file system specification, URL or database key, constitutes the value for the I_IMAGE and I_THUMBNAIL fields in the ITEM table. In any event, the storage on the SUT will include the capacity necessary for all the objects. The population of these fields is implementation specific.

4.6.2.13 The customer email address field (C_EMAIL) is generated by the concatenation of the corresponding value in C_UNAME followed by the special character "@" followed by a random a-string [2 .. 9] followed by the string of characters ".com".

Example: Given a C_UNAME of RIULOGNG, C_EMAIL, the customer email address, may be RIULOGNG@bjs2acKd.com.

4.6.2.14 The country name (**CO_NAME**) must be chosen at random from the following list, shown here along with their associated exchange rate, **CO_EXCHANGE**, and currency, **CO_CURRENCY**:

Name	Exchange	Currency	Name	Exchange	Currency
United States	1	Dollars	United Kingdom	0.625461	Pounds
Canada	1.46712	Dollars	Germany	1.86125	Deutsche Marks
France	6.24238	Francs	Japan	121.907	Yen
Netherlands	2.09715	Guilders	Italy	1842.64	Lira
Switzerland	1.51645	Francs	Australia	1.54208	Dollars
Algeria	65.3851	Dinars	Argentina	0.998	Pesos
Armenia	540.92	Dram	Austria	13.0949	Schillings
Azerbaijan	3977	Manat	Bahamas	1	Dollars
Bahrain	0.3757	Dinar	Bangla Desh	48.65	Taka
Barbados	2	Dollars	Belarus	248000	Rouble
Belgium	38.3892	Francs	Bermuda	1	Dollars
Bolivia	5.74	Boliviano	Botswana	4.7304	Pula
Brazil	1.71	Real	Bulgaria	1846	Lev
Cayman Islands	0.8282	Dollars	Chad	627.1999	Franc
Chile	494.2	Pesos	China	8.278	Yuan Renmimbi
Christmas Island	1.5391	Dollars	Colombia	1677	Pesos
Croatia	7.3044	Kuna	Cuba	23	Pesos
Cyprus	0.543	Pounds	Czech Republic	36.0127	Koruna
Denmark	7.0707	Kroner	Dominican Republic	15.8	Pesos
Eastern Caribbean	2.7	Dollars	Ecuador	9600	Sucre
Egypt	3.33771	Pounds	El Salvador	8.7	Colon
Estonia	14.9912	Kroon	Ethiopia	7.7	Birr
Falkland Island	0.6255	Pound	Faroe Island	7.124	Krone
Fiji	1.9724	Dollars	Finland	5.65822	Markka
Gabon	627.1999	Franc	Gibraltar	0.6255	Pound
Greece	309.214	Drachmas	Guam	1	Dollars
Hong Kong	7.75473	Dollars	Hungary	237.23	Forint
Iceland	74.147	Krona	India	42.75	Rupees
Indonesia	8100	Rupiah	Iran	3000	Rial
Iraq	0.3083	Dinar	Ireland	0.749481	Punt
Israel	4.12	Shekels	Jamaica	37.4	Dollars
Jordan	0.708	Dinar	Kazakhstan	150	Tenge
Kuwait	0.3062	Dinar	Lebanon	1502	Pounds
Luxembourg	38.3892	Francs	Malaysia	3.8	Ringgit
Mexico	9.6287	Pesos	Mauritius	25.245	Rupees
New Zealand	1.87539	Dollars	Norway	7.83101	Kroner
Pakistan	52	Rupees	Philippines	37.8501	Pesos
Poland	3.9525	Zloty	Portugal	190.788	Escudo
Romania	15180.2	Leu	Russia	24.43	Rubles

Saudi Arabia	3.7501	Riyal	Singapore	1.72929	Dollars
Slovakia	43.9642	Koruna	South Africa	6.25845	Rand
South Korea	1190.15	Won	Spain	158.34	Pesetas
Sudan	5.282	Dinar	Sweden	8.54477	Krona
Taiwan	32.77	Dollars	Thailand	37.1414	Baht
Trinidad	6.1764	Dollars	Turkey	401500	Lira
Venezuela	596	Bolivar	Zambia	2447.7	Kwacha

4.6.2.15 The item backing (**I_BACKING**) must be chosen at random from the following list:

- HARDBACK
- PAPERBACK
- USED
- AUDIO
- LIMITED-EDITION

4.6.2.16 The item dimensions (**I_DIMENSIONS**) must be generated by the concatenation of 3 numeric, 5.2 digits separated by an "x".

Example: 12.25x16.50x1.25

4.6.2.17 The item title (**I_TITLE**) must be generated as a random a-string [14..60]. It must have an embedded substring generated as DigSyl(num, 7). The substring must start at an offset within the I_TITLE string random within [0 .. 46].

```
if A_ID <= (NUM_ITEMS / 5)
        num = I_ID;
else
        num = random within [0 .. (NUM_ITEMS / 5)];
```

Example: Given an I_ID of 4628, the embedded substring is generated as DigSyl(4628, 7). Thus, the I_TITLE corresponding to an I_ID of 4628 would contain the substring BABABAREATALIN starting at some random offset between 0 and 46.

Comment: The intent of this Clause is to ensure that web interactions that execute a search based on the content of I_TITLE find a number of matching records.

4.6.2.18 The author last name (**A_LNAME**) must be generated as a random a-string [14..20] It must have an embedded substring generated as DigSyl(num, 7). The substring must start at an offset within the A_LNAME string random within [0..6].

```
if A_ID <= (NUM_AUTHORS / 2.5)
        num = I_ID;
else
        num = random within [0 .. (NUM_AUTHORS / 2.5)];
```

Example: Given an A_ID of 543, the embedded substring is generated as DigSyl(543, 7). Thus, the A_LNAME corresponding to an A_ID of 543 would contain the substring BABABABASERERI starting at some random offset between 0 and 6.

Comment: The intent of this Clause is to ensure that web interactions that execute a search based on the content of A_LNAME find a number of matching records.

4.7 Table Population Requirements

4.7.1 The initial database population must be comprised of the following (where NUM_ITEMS is the number of records in the Item table, NUM_ORDERS in the Order table, NUM_AUTHORS in the Author table and NUM_CUSTOMERS in the Customer table):

- [1000, 10,000, 100,000, 1,000,000, 10,000,000] rows in the ITEM table (depending on the scale factor chosen) with:
  - I_ID unique within [1 .. NUM_ITEMS]
  - I_TITLE generated according to Clause 4.6.2.17
  - I_A_ID random within [1 .. (NUM_ITEMS / 4)]
  - I_PUB_DATE random date between January 1, 1930 and current date
  - I_PUBLISHER random a-string [14 .. 60]
  - I_SUBJECT generated according to Clause 4.4.2.11
  - I_DESC random a-string [100 .. 500]
  - I_RELATED1 random I_ID
  - I_RELATED2 random I_ID
  - I_RELATED3 random I_ID
  - I_RELATED4 random I_ID
  - I_RELATED5 random I_ID
  - I_THUMBNAIL, graphic object, generated according to Clause 4.6.2.12
  - I_IMAGE, graphic object, generated according to Clause 4.6.2.12
  - I_SRP random within [1.00 .. 9,999.99]
  - I_COST generated as I_SRP – (random within [(0 .. 0.5) * I_SRP])
  - I_AVAIL generated as I_PUB_DATE + (random within [1 .. 30] days)
  - I_ISBN random a-string of 13 characters
  - I_PAGE random within [20 .. 99,999]
  - I_BACKING, variable size text, generated according to Clause 4.4.2.15
  - I_DIMENSIONS (length x width x height of the book), generated according to Clause 4.6.2.16
- 92 rows in the COUNTRY table with:
  - CO_ID unique within [1 .. 92]
  - CO_NAME selected from the table in Clause 4.6.2.14

- CO_EXCHANGE selected from the table in Clause 4.6.2.13
- CO_CURRENCY selected from the table in Clause 4.6.2.13

♦ (NUM_ITEMS / 4) rows in the AUTHOR table with:

- A_ID unique within [1 .. (NUM_ITEMS / 4)]
- A_FNAME random a-string [3 .. 20]
- A_MNAME random a-string [1 .. 20]
- A_LNAME generated according to Clause 4.6.2.18
- A_DOB random date between January 1, 1800 and January 1, 1990
- A_BIO random a-string [125 .. 500]

♦ (2880 * # EBs) rows in the CUSTOMER table with :

- C_ID unique within [1 .. 2880 * # EBs]
- C_UNAME generated according to Clause 4.6.2.9
- C_PASSWD generated according to Clause 4.6.2.10
- C_LNAME random a-string [8 .. 15]
- C_FNAME random a-string [8 .. 15]
- C_ADDR_ID random within [1 .. 2 * 2880 * # EBs]
- C_PHONE random n-string [9 .. 16]
- C_EMAIL generated according to Clause4.6.2.13
- C_SINCE generated as current date – random within [1 .. 730] days
- C_LAST_VISIT generated as C_SINCE + random within [0 .. 60] days, but not exceeding the current date
- C_LOGIN date / time given by the operating system when the table was populated
- C_EXPIRATION generated as C_LOGIN + 2 hours
- C_DISCOUNT random within [0.00 .. 0.50]
- C_BALANCE = 0.00
- C_YTD_PMT random within [0.00 .. 999.99]
- C_BIRTHDATE random date between January 1, 1880 and current date
- C_DATA random a-string [100 .. 500]

♦ (0.9 * NUM_CUSTOMERS) rows in the ORDERS table with:

- O_ID unique within [1 .. (0.9 * NUM_CUSTOMERS)]
- O_C_ID random within [1 .. NUM_CUSTOMERS]
- O_DATE generated as current date – random within [1 .. 60] days
- O_SUB_TOTAL random within [10.00 .. 9999.99]
- O_TAX generated as O_SUB_TOTAL * 0.0825
- O_TOTAL generated as O_SUB_TOTAL + O_TAX + 3.00 + (1.00 * count_of_items_in_order)

- O_SHIP_TYPE selected at random from the following:

  AIR, UPS, FEDEX, SHIP, COURIER, MAIL
- O_SHIP_DATE generated as O_DATE + random within [0 .. 7] days
- O_BILL_ADDR_ID = C_ADDR_ID
- O_SHIP_ADDR_ID = random within [1 .. (2 * NUM_CUSTOMERS)]
- O_STATUS selected at random from the following:

  PROCESSING, SHIPPED, PENDING, DENIED

♦ (3 * NUM_ORDERS) rows in the ORDER_LINE table with:

- OL_ID unique within [1 .. 100]
- OL_O_ID random within [1 .. NUM_ORDERS]
- OL_I_ID random within [1 .. NUM_ITEMS]
- OL_QTY random within [1 .. 300]
- OL_DISCOUNT random within [0.00 .. 0.03]
- OL_COMMENTS random a-string [20 .. 100]

♦ (1 * NUM_ORDERS) rows in the CC_XACTS table with:

- CX_O_ID = O_ID
- CX_TYPE selected at random from the following:

  VISA, MASTERCARD, DISCOVER, AMEX, DINERS
- CX_NUM random n-string of 16 digits
- CX_NAME random a-string [14 .. 30]
- CX_EXPIRY current date + random within [10 .. 730] days
- CX_AUTH_ID random a-string of 15 characters
- CX_XACT_AMT = O_TOTAL
- CX_XACT_DATE = O_SHIP_DATE
- CX_CO_ID random within [1 .. 92]

♦ (2 * NUM_CUSTOMERS) rows in the ADDRESS table with:

- ADDR_ID unique within [1 .. (2 * NUM_CUSTOMERS)]
- ADDR_STREET1 random a-string [15 .. 40]
- ADDR_STREET2 random a-string [15 .. 40]
- ADDR_CITY random a-string [4 .. 30]
- ADDR_STATE random a-string [2 .. 20]
- ADDR_ZIP random a-string [5 .. 10]
- ADDR_CO_ID random within [1 .. 92]

4.7.2 The implementation may not take advantage of the fact that some fields are initially populated with a fixed value. For example, storage space cannot be saved by defining a default value for any particular field and storing this value only once in the database.

Clause 5 - PERFORMANCE METRICS AND RESPONSE TIME

The two primary metrics of the TPC-W benchmark are the number of Web Interactions Per Second (WIPS), and a price performance metric defined as Dollars/WIPS ($/WIPS). The web interaction mix used for the primary metrics are intended to reflect an average shopping scenario. It is understood that some shopping applications attract users primarily interested in browsing, while others attract those planning to purchase. To provide insight as to how a particular system will perform under these conditions, two additional metrics, namely the WIPSb and the WIPSo, have been defined.

5.1 Definition of Terms

5.1.1 The term Measurement Interval is used in this specification to refer to a steady state period during the execution of the benchmark for which the test sponsor is reporting a performance metric (see Clause 5.5 for detailed requirements.)

5.1.2 The term Shopping Interval is used in this specification to refer to the reported Measurement Interval (see Clause 5.5) during the execution of a shopping scenario composed of a mix of browsing and ordering activity.

5.1.3 The term Browsing Interval is used in this specification to refer to the reported Measurement Interval (see Clause 5.5) during the execution of a browsing scenario composed of significant browsing activity and relatively little ordering activity.

5.1.4 The term Ordering interval is used in this specification to refer to the reported Measurement Interval (see Clause 5.5) during the execution of an ordering scenario composed of significant ordering activity and relatively little browsing activity.

5.1.5 The term Web Interaction Response Time (WIRT) is used in this specification to refer to the time elapsed from the first byte sent by the EB to request a web interaction until the last byte received by the EB to complete the web interaction (see Clause 5.3 for detailed requirements.)

5.1.6 The term **Think Time** is used in this specification to refer to the time elapsed from the last byte received by the EB to complete a web interaction until the first byte sent by the EB to request the next web interaction (see Clause 5.3 for detailed requirements.)

5.1.7 The term WIPS is used in this specification to refer to the average number of Web Interactions Per Second completed during the Shopping Interval.

5.1.8 The term $/WIPS is used in this specification to refer to the total cost of the SUT (see Clause 7) divided by the number of WIPS measured during the Shopping Interval.

5.1.9 The term WIPSb is used in this specification to refer to the average number of Web Interactions Per Second completed during the Browsing Interval. (WIPSb is a secondary metric.)

5.1.10 The term WIPSo is used in this specification to refer to the average number of Web Interactions Per Second completed during the Ordering Interval. (WIPSo is a secondary metric.)

5.2 Web Interaction Mixes

The TPC-W workload is made up of a set of web interactions specified in details in Clause 2. In general, these web interactions can be classified as either “Browse” or “Order” depending on whether they involve browsing and searching on the site or whether they play an explicit role in the ordering process. The categorization used for this benchmark is found in the table in Clause 5.2.1.

The benchmark defines three distinct Measurement Intervals, each for a distinct mix of web interactions, resulting in three distinct performance metrics. WIPS is the primary performance metric as it is based on an average shopping scenario. WIPSb and WIPSo are secondary performance metrics. While similar to the WIPS metric, the web interaction mix for WIPSb is primarily browsing, and the web interaction mix for WIPSo involves a significant percentage of ordering resulting in transaction processing. The characterization of the various web interaction mixes are found in Clause 5.2.1 below.

5.2.1 Web Interaction Mix Requirements

Over each Measurement Interval, the RBE must maintain the mix of web interactions specified in the table below**.** The RBE is expected to deviate from the specified percentages as a natural result of the finite random selection process. The maximum deviation allowed is (0.05 x P) where P is any of the required mix percentages.

Example: If the Home web interaction is 16.00% (as in the WIPS mix), it is required that the average mix percentage of Home during the Measurement Interval falls between 16.80% and 15.20%.

Comment: For measurements resulting in a reported throughput of less than 500WIPS, the maximum deviation allowed for the Admin Request and Admin Confirm web interaction is (0.1 x P).

Web Interaction	Browsing Mix (WIPSb)	Shopping Mix (WIPS)	Ordering Mix (WIPSo)
Browse	**95 %**	**80 %**	**50 %**
Home	29.00 %	16.00 %	9.12 %
New Products	11.00 %	5.00 %	0.46 %
Best Sellers	11.00 %	5.00 %	0.46 %
Product Detail	21.00 %	17.00 %	12.35 %
Search Request	12.00 %	20.00 %	14.54 %
Search Results	11.00 %	17.00 %	13.08 %
Order	**5 %**	**20 %**	**50 %**
Shopping Cart	2.00 %	11.60 %	13.53 %
Customer Registration	0.82 %	3.00 %	12.86 %
Buy Request	0.75 %	2.60 %	12.73 %
Buy Confirm	0.69 %	1.20 %	10.18 %
Order Inquiry	0.30 %	0.75 %	0.25 %
Order Display	0.25 %	0.66 %	0.22 %
Admin Request	0.10 %	0.10 %	0.12 %
Admin Confirm	0.09 %	0.09 %	0.11 %

5.2.2 Maintaining the Web Interaction Mix

5.2.2.1 During the Measurement Interval, each EB cycles through a process of requesting a web interaction, receiving and parsing the response page, measuring the time required to receive it (WIRT), selecting the next navigation option, preparing the next request and waiting the balance of Think Time before requesting the next web interaction. The required mix of web interactions is maintained by controlling the selection of the next navigation option. This clause specifies how this control must be implemented by the RBE.

5.2.2.2 Each web interaction results in the presentation of a page to the EB. After receiving that page, the EB selects the web interaction to be requested next. Every valid selection is one of the set of navigation options permitted for the current web interaction as specified in Clause 2. Having received the response page for a particular web interaction, the EB selects the next web interaction using the method described in Clause 5.2.2.3, which is guaranteed to satisfy the web interaction mix requirements defined in Clause 5.2.1.

5.2.2.3 Each web interaction has a set of navigation options. An integer between 1 and 9,999, called a **Threshold**, is assigned to each navigation option. To select its next request the EB generates a random number, with at least 31 bits of precision, from a uniform distribution between 1 and 9,999. It then selects the navigation option for which the Threshold is equal to or most immediately greater than the random number.

5.2.2.4 The following tables list the Thresholds that must be used by the EB to select each navigation option during each of the three Measurement Intervals:

Thresholds for the Shopping Interval (WIPS)

To this Web Interaction → / From this Response Page ↓	Admin Confirm	Admin Request	Best Sellers	Buy Confirm	Buy Request	Customer Regist.	Home	New Products	Order Display	Order Inquiry	Product Detail	Search Request	Search Results	Shopping Cart
Admin Confirm							9952					9999		
Admin Request	8999						9999							
Best Sellers							167				472	9927		9999
Buy Confirm							84					9999		
Buy Request				4614			6546							9999
Customer Regist.					8666		8760					9999		
Home			3124					6249		6718		7026		9999
New Products							156				9735	9784		9999
Order Display							69					9999		
Order Inquiry							72		8872			9999		
Product Detail		58					832				1288	8603		9999
Search Request							635						9135	9999
Search Results							2657				9294	9304		9999
Shopping Cart						2585	9552							9999

Thresholds for the Browsing Interval (WIPSb)

To this Web Interaction → / From this Response Page ↓	Admin Confirm	Admin Request	Best Sellers	Buy Confirm	Buy Request	Customer Regist.	Home	New Products	Order Display	Order Inquiry	Product Detail	Search Request	Search Results	Shopping Cart
Admin Confirm							9877					9999		
Admin Request	8999						9999							
Best Sellers							4607				5259	9942		9999
Buy Confirm							342					9999		
Buy Request				9199			9595							9999
Customer Regist.					9145		9619					9999		
Home			3792					7585		7688		9559		9998

New Products							299				9867	9941		9999
Order Display							802					9999		
Order Inquiry							523		8856			9999		
Product Detail		47					8346				9749	9890		9999
Search Request							788						9955	9999
Search Results							3674				9868	9942		9999
Shopping Cart						4099	8883							9999

Thresholds for the Ordering Interval (WIPSo)

To this Web Interaction → From this Response Page ↓	**Admin Confirm**	**Admin Request**	**Best Sellers**	**Buy Confirm**	**Buy Request**	**Customer Regist.**	**Home**	**New Products**	**Order Display**	**Order Inquiry**	**Product Detail**	**Search Request**	**Search Results**	**Shopping Cart**
Admin Confirm							8348					9999		
Admin Request	8999						9999							
Best Sellers							1				333	9998		9999
Buy Confirm							2					9999		
Buy Request				7999			9453							9999
Customer Regist.					9899		9901					9999		
Home			499					999		1269		1295		9999
New Products							504				9942	9976		9999
Order Display							9939					9999		
Order Inquiry							1168		9968			9999		
Product Detail		99					3750				5621	6341		9999
Search Request							815						9815	9999
Search Results							486				7817	9998		9999
Shopping Cart						9499	9918							9999

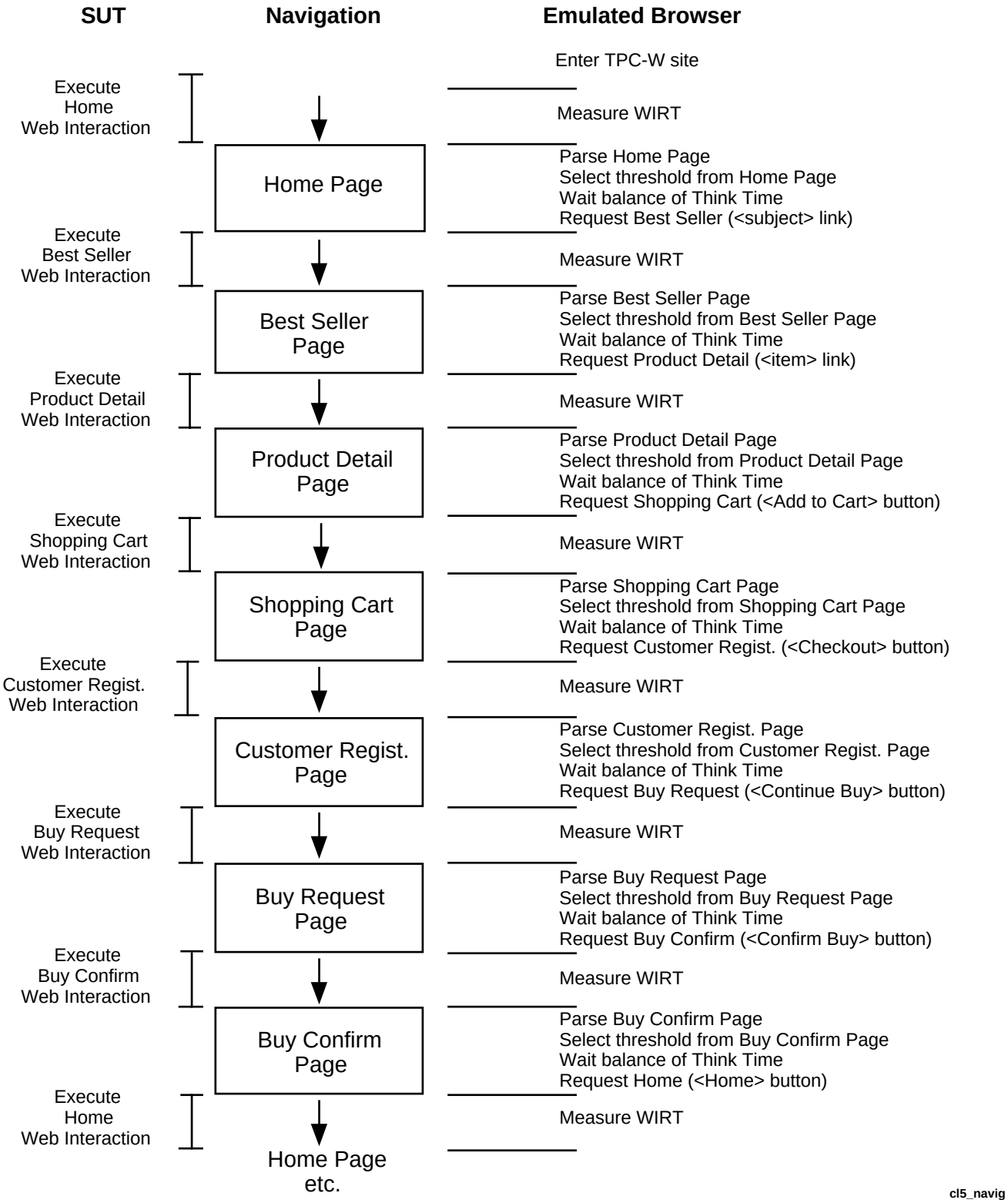
Comment 1: Following is an example of how the above tables should be used during a Shopping Interval. After completing a Home web interaction the EB generates the random number 6371. No Threshold is equal to that random number and the Threshold that is most immediately greater than 6371 is 6718. This indicates that an Order Inquiry web interaction must be selected.

Comment 2: Many cells of the above tables contain no Threshold. This means that no navigation option exists to move in that direction. One may not, for example, select the Buy Request web interaction from the Home Page.

Comment 3: Each EB must use the Thresholds specified in the above tables, even though other Thresholds exist that could be used to also conform to the table in Clause 5.2.1 and to obtain the same overall mix of web interactions.

5.2.3 Example of EB Activity on the TPC-W Site

The following figure illustrates an example of how an EB might move through the TPC-W web site:



5.3 Think Time and Response Time

5.3.1 Think Time

5.3.2 The Think Time (TT) is defined by:

TT = T2 - T1

where:

T1 and T2 are measured at the EB;

T1 = time measured after the last byte of the last response page is received by the EB from the SUT; and

T2 = time measured before the first byte of the next request is sent by the EB to the SUT.

Comment 1: The resolution of the time stamps must be at least 0.1 seconds.

5.3.2.1 For each type of web interaction, the Think Time must be taken independently from a negative exponential distribution. Each Think Time, T_t, must be computed from the following equation:

$T_t = - \ln(r) * \mu$

where:

ln = natural log (base e)

r = random number uniformly distributed between 0 and 1.0

μ = mean think time of 7 seconds

Comment 1: The resolution of the time stamps must be at least 0.1 seconds.

Comment 2: Each distribution must be truncated at 10 times its mean value (i.e., truncated at 70 seconds, inclusif).

5.3.2.2 The average duration of the Think Time over the Measurement Interval, for each type of web interactions, must be no less than seven (7) seconds.

5.3.2.3 It is expected that the EB does some amount of processing during the Think Time (e.g., parsing the response page, generating the next request, logging statistics, etc.) As a result, the amount of time that the EB must wait after completing all processing and before sending the first byte of the next request to the SUT can be reduced to the balance of the computed Think Time for that web interaction.

5.3.3 Web Interaction Response Time (WIRT)

5.3.4 The Web Interaction Response Time (WIRT) is defined by:

WIRT = T2 - T1

where:

T1 and T2 are measured at the EB;

T1 = time measured before the first byte of the first HTTP request of the web interactionis sent by the EB to the SUT; and

T2 = time measured after the last byte of the last HTTP response that completes the web interactionis received by the EB from the SUT.

Comment 1: The resolution of the time stamps must be at least 0.1 seconds.

Comment 2: Although the intent of the benchmark is to measure response time as experienced by an actual user, the time needed for the browser to display the web pages and objects is not included in the WIRT.

5.3.4.1 During the Measurement Intervals, at least 90% of web interactions of each type must have a WIRT of less than the constraint specified for that web interaction in the table below. For example, at least 90% of all Home web interactions must have a WIRT of less than 1 second.

	Admin Confirm	Admin Request	Best Sellers	Buy Confirm	Buy Request	Customer Regist.	Home	New Products	Order Display	Order Inquiry	Product Detail	Search Request	Search Results	Shopping Cart
90% WIRT Constraint	20	1	5	5	1	2	1	5	2	1	1	1	10	1

5.3.4.2 Over the Measurement Interval, the 90th percentile WIRT for each type of web interaction must be no less than 0.1 second shorter than the average WIRT for that web interaction.

5.4 Computation of Throughput Rating

5.4.1 The throughput for each of the three Measurement Intervals is computed as the total number of web interactions requested and completed within that Measurement Interval divided by the length of that Measurement Interval in seconds.

Comment: Web interactions can be included in the computation of the reported throughput only if their WIRT is fully within the bounds of the Measurement Interval.

5.4.2 All reported throughputs must be measured, rather than interpolated or extrapolated, and expressed to exactly two decimal places, rounded down to the hundredth place. For example, suppose 105.548 WIPS is measured on test for which all 90% WIRT constraints are met and 117.572 WIPS is measured on test for which some 90% WIRT constraints are exceeded. Then the reported WIPS is 105.54 rather than some interpolated value between 105.548 and 117.572.

5.5 Measurement Interval Requirements

This Clause defines the requirements for the three Measurement Intervals - one for Shopping, one for Browsing and one for Ordering.

5.5.1 Test Run

5.5.1.1 A Test Run is made of one or more Measurement Intervals. Each Measurement Interval must be preceded by a ramp-up period of at least 10 minutes at the end of which a steady state throughput level must be reached. Immediately following the last Measurement Interval in a Test Run, the steady state throughput level of that Measurement Interval must be maintained for at least 5 minutes..

5.5.1.2 The three Measurement Intervals can be executed either as three individual Test Runs, or as one or two Test Runs where two or more Measurement Intervals are executed serially within the same Test Run.

5.5.1.3 If more than one Measurement Interval is executed within the same Test Run, the mix of web interactions must be adjusted during the ramp-up period between the Measurement Intervals.

5.5.2 Steady State

5.5.2.1 The reported throughput must be computed over a Measurement Interval during which the throughput level is in a steady state condition that represents the true sustainable performance of the SUT.

5.5.2.2 Although each Measurement Interval may be as short as 30 minutes, the SUT must be configured so that it is possible to run the mix of web interactions during 14 days of uninterrupted execution (as defined in Clause 6.6.1) while maintaining full ACID properties.

Comment 1: For example, the media used to store the database log data until it can be archived without interruption of processing must be configured if required to recover from any single point of failure

Comment 2: An example of a configuration that would not comply is one where a log file is allocated such that better performance is achieved during the measured portion of the run than during the remaining portion of any full throughput period, perhaps because a dedicated device was used initially but space on a shared device is used later in the full throughput period.

Comment 3: Steady state is easy to define (e.g., sustainable throughput) but difficult to prove. The test sponsor (and/or the auditor) is required to report the method used to verify steady state sustainable throughput and the reproducibility of measured results. The auditor is encouraged to use available monitoring tools to help determine the steady state.

Comment 4: An example of compliant implementation would be one where the entire database is placed on redundant storage and the database log data is stored on a circular files large enough to span an complete checkpoint cycle.

5.5.2.3 Some aspects of the benchmark implementation can result in systematic but small variations in sustained throughput over an 8 hour period. These variations must be limited to 5% of the reported throughput. This implies that no interval of a duration equal to the Measurement Interval (with a granularity of 30 seconds over 8 hours) can have a computed throughput that varies from the reported throughput by more than 5%.

Comment: There is no specific requirement to execute an 8 hour Test Run, but the auditor may find it necessary to execute up to a full 8 hour Test Run to demonstrate sustained throughput.

5.5.2.4 To prevent significant alteration to the properly scaled database population, the mix of web interactions and the requirements summarized in Clause 5.5 must be followed during the entire Test Run as well as during any other time between Test Runs.

5.5.2.5 To demonstrate the reproducibility of the steady state condition during the Shopping Interval, a minimum of one additional (and non-overlapping) Shopping Interval of the same duration as the reported Shopping Interval must be measured and its WIPS must be disclosed and be equal to or greater than the reported WIPS. The reproducibility test is not required for either the Browsing Interval or the Ordering Interval.

5.5.3 Duration

5.5.3.1 Each Measurement Interval must:

- Begin after the system has reached steady state with the corresponding mix of web interactions.
- Be long enough to generate reproducible throughput results which are representative of the performance which would be achieved during a sustained 8 hour period.
- Extend uninterrupted for a minimum of 30 minutes.

5.5.3.2 Some systems do not write modified database records/pages to durable media at the time of modification, but instead defer these writes. At some subsequent time, the modified records/pages are written to make the durable copy current. This process is defined as a checkpoint in this document.

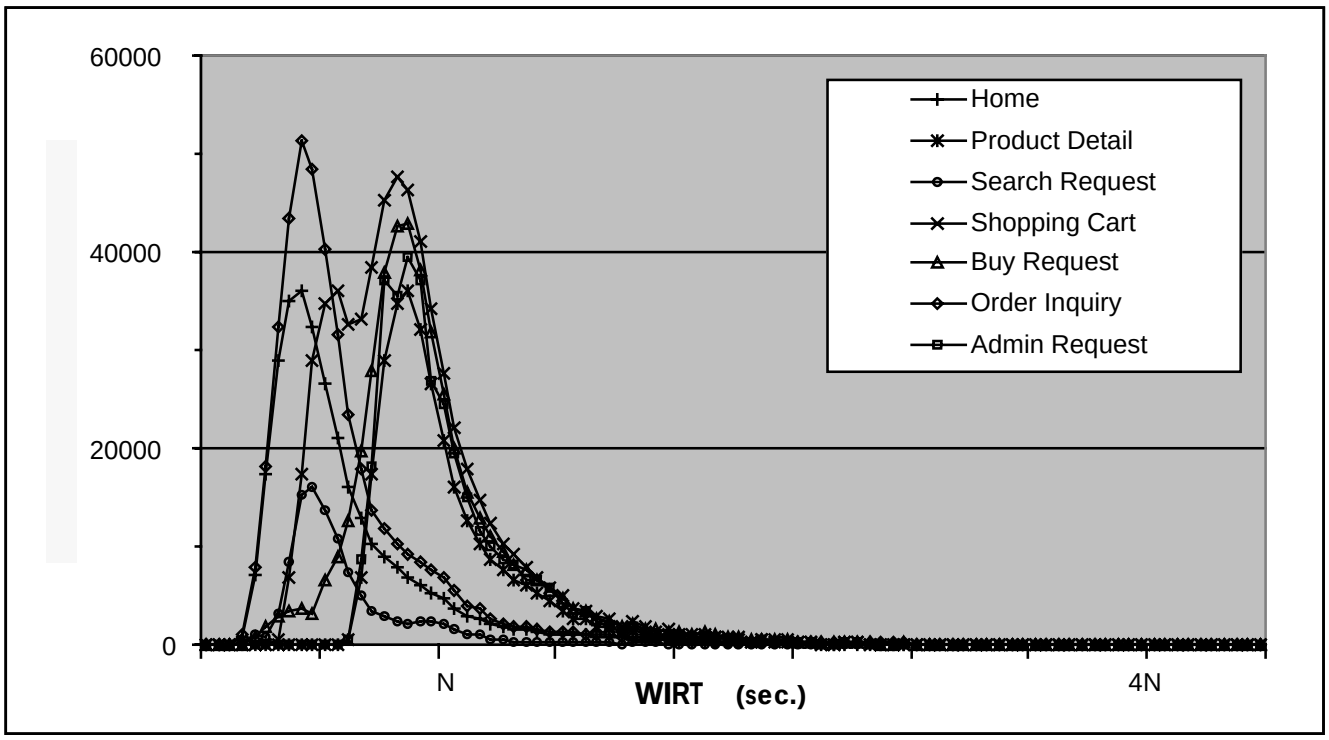
5.5.3.3 It is a requirement that no recovery data older than 15 minutes prior to an instantaneous interruption be used for recovery. The consequence of this requirement is that the database contents stored on durable media cannot at any time during the Test Run be more than 15 minutes older than the most current state of the database (±5%).

5.5.3.4 All work required to perform a checkpoint must occur in full at least once before the Measurement Interval begins but after steady state has been reached.

Comment: Regardless of the checkpointing method used (e.g., discrete checkpoints, continuous checkpoints, trickle checkpoints) a full cycle of checkpointing must be completed before each Measurement Interval.

5.6 Required Reporting

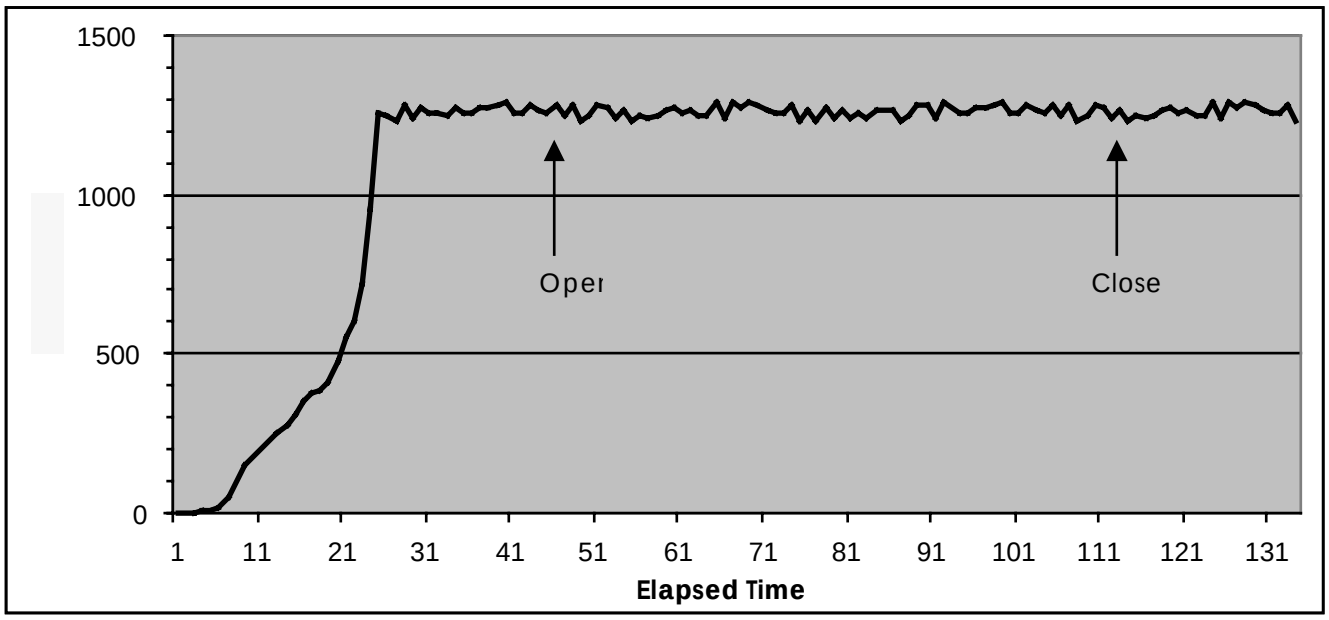
5.6.1 The frequency distribution of WIRT of all web interactions, started and completed during the Browsing Interval, the Shopping Interval and the Ordering Interval, must be reported in a graphical format. Each graph must represent the distribution of seven (7) web interactions for each Measurement Interval, for a total of 6 graphs. In each graph, the x-axis represents the WIRT and must range from 0 to four times the longest required 90th percentile WIRT (N) of the seven web interactions. This 0 to 4N WIRT range is to be divided into 100 equal length intervals. One additional interval will include the WIRT range from 4N to infinity. All 101 intervals must be reported. The y-axis represents the frequency of each of the seven web interactions at a given WIRT range with a granularity of at least 10 intervals. An example of such a graph is shown below:



For each Measurement Interval, the web interactions must be gouped as follows on the two graphs of WIRT frequency distribution:

- First Graph: Home, Product Detail, Search Request, Shopping Cart, Buy Request, Order Inquiry and Admin Request.
- Second Graph: New Products, Best Sellers, Search Results, Customer Registration, Buy Confirm, Order Display and Admin Confirm.

5.6.2 For all three Measurement Intervals, a graph of the measured throughput versus elapsed time (i.e., wall clock time) must be reported for the entire Test Run. The x-axis represents the elapsed time from the start of the run. The y-axis represents the throughput in WIPS. At least 60 different intervals must be used with a maximum interval size of 30 seconds. The opening and the closing of the Measurement Interval must also be reported. An example of such a graph is shown below.



5.7 Additional Performance Metrics

In order to provide additional information to the benchmark customer, the TPC-W workload requires the disclosure of a small set of performance metrics that are available from common commercial system monitor facilities. Examples of these facilities are the UNIX-based sar measurement facilities, the OS/390-based RMF facilities, the Windows NT performance monitor, and the commonly available set of DBMS monitoring facilities built-in to each database vendor's products.

The objective in providing this performance monitor data in the Full Disclosure Report is to enhance the customer's ability to fully understand and interpret the results of the benchmark measurement. In this way the user of the data will have a much higher level of confidence in the reported measurement and will be better able to gain the necessary insight to properly interpret the results. Also, the data will help the user apply the benchmark results to his own application environment because the detailed performance data can be used to identify similarities and dissimilarities between the customer's environment and the vendor's benchmark.

There are two classes of performance metrics:

- Those that are required, in that they are available in one form or another from all current computer systems.

- Those that are optional, in that they may provide a level of detail unique to a specific vendor or architecture that promotes a better understanding of the vendor's system. This broader set of optional data is useful but is not required. Optional data should be reported such that it's meaning and interpretation are the same as provided by the vendor's predominant standard performance monitor.

A careful definition of the data fields reported must be provided to allow the data to be understood and properly interpreted. It is anticipated that most of the data definition and descriptive information will be taken from the vendor's product documentation.

Note: Performance monitor data is used to promote a more complete understanding of individual platform results, and is not for direct comparison between different platforms.

5.7.1 General Guidelines

5.7.1.1 The set of required performance metrics, as specified in Clause 5.7.2, must be disclosed for the Shopping, Browsing, and Ordering Intervals using the format specified in Appendix G.

5.7.1.2 Data for these reported metrics must be collected during the same Measurement Intervals from which the reported **WIPS**, **WIPSb, and WIPSo** rates are produced.

Comment: The intent of this clause recognizes that monitoring facilities may generate measurable amounts of overhead during operation. This clause also recognizes that these monitor facilities are usually present in customer environments, and that the benchmark performance indicated by the TPC-W benchmark should reflect their presence.

5.7.1.3 The monitoring tools used to provide these performance metrics must be commercially available, reported and priced.

5.7.1.4 Implementations that involve multiple systems (e.g., multi-tier configurations) must report a separate set of performance metrics for each system. In a group of systems performing the same processing function, if each required performance monitor metric falls within 5% across all the systems in the group, they then can each be reported as an aggregated metric over all the systems in the group.

Comment: In this clause a system is defined as running a single copy of the operating system sharing a single memory space.

5.7.1.5 The operational method used to obtain monitor data must be reported. For example, that the monitor was invoked as a started task, and that binary data was recorded to an external disk or tape file for post-processing.

5.7.1.6 If a sampling technique is used to obtain performance data then the sample rate (per second) must be reported.

Comment: The intent of this clause is to report the sample rate for non-event-driven system monitor facilities. An example of this sample rate is found in determining "channel-busy", where the status of the channel must be sampled in order to determine if the resource is indeed busy.

5.7.1.7 If there is no relevance or meaning to a specific data metric for the system and products used in this specific benchmark implementation, then the symbol (N/A) for not applicable may be used. This exclusion may not be used to avoid reporting relevant but uncollected data.

5.7.2 Required Performance Monitor Metrics

This clause describes the set of required performance metrics. Refer to the table in Appendix G for specific details of reporting layouts and accuracy.

If a multi-tier system is not used, the metrics are reported on an aggregate basis across all host systems. If the system uses multiple tiers of SUT functionality, the metrics must be reported on each tier.

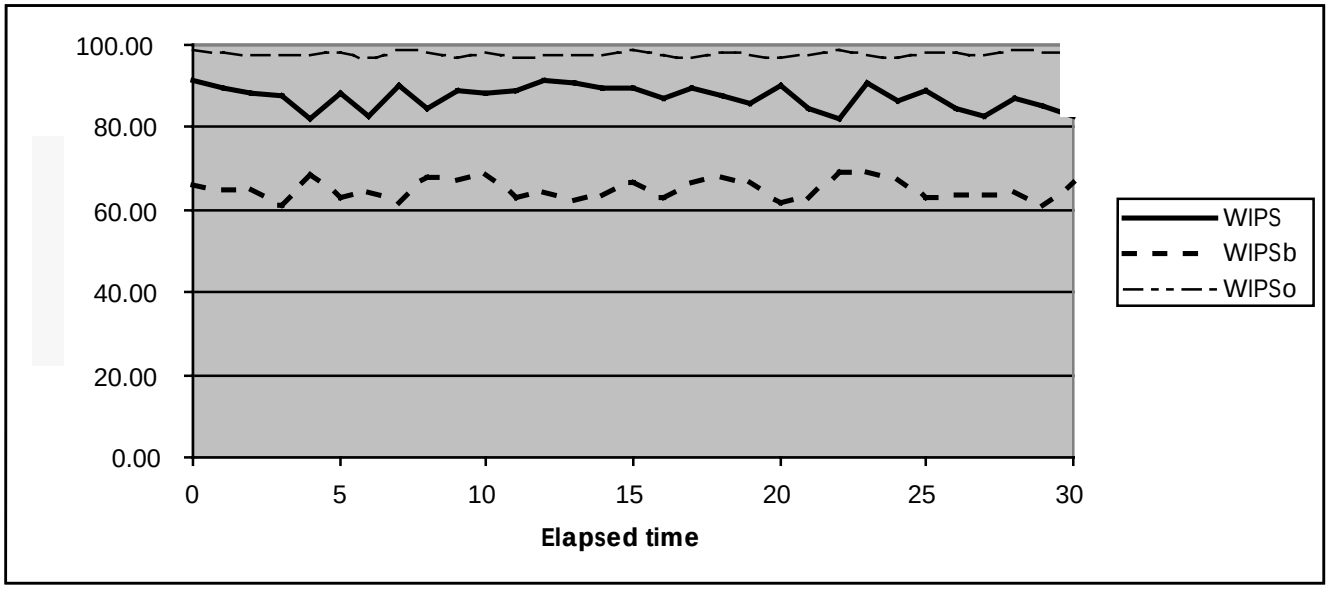
5.7.2.1 CPU Utilization

A key parameter of any capacity measurement is CPU utilization. All enterprise-class system monitoring facilities provide a means of recording CPU utilization.

5.7.2.1.1 The required reporting method is to record CPU utilization, i.e. the percentage of time for which the CPU was busy.

5.7.2.1.2 For each Measurement Interval (WIPS, WIPSo and WIPSb), at least one graph illustrating CPU utilization must be reported. The x-axis represents the elapsed time from the start of the Measurement Interval. The y-axis represents the average CPU utilization on a scale of 0 to 100%. Discrete one minute intervals must be graphed across the total Measurement Interval, where each value is the average CPU utilization over that one minute interval. The three Measurement Intervals can be combined on a single graph.

An example of such a graph is shown below:



5.7.2.2 Memory Utilization

The manner in which memory is allocated and controlled by the host operating system can indicate how efficiently the vendor operating system and/or DBMS supports the demands of the TPC-W benchmark. To further this understanding:

5.7.2.2.1 The total amount of physical memory available to the SUT during the Measurement Interval must be reported.

5.7.2.2.2 The total amount of dedicated database "pool", commonly known as shared memory, must be reported. In implementations where multiple pools or processing regions are used for different database elements, only the aggregate sum of all such pools need be reported.

5.7.2.2.3 The memory system page size must be reported.

Comment: It is expected that the data provided will be identical to that which is reported by the standard performance monitor used.

5.7.2.3 Page/Swap Activity

The average Page-in, Page-out, Swap-in, and Swap-out rates (where applicable) are to be reported.

5.7.2.4 Database I/O Activity

There are four required rates (reported as rates per second). They are Logical READS, Physical READS, Logical WRITES, and Physical WRITES. The standard measures of Logical READ/WRITE for a particular system should be used along with a sufficiently detailed definition to allow proper interpretation.

Comment: The objective is to provide some view of how efficiently the system under test has cached data to satisfy upcoming application data needs.

5.7.2.5 System I/O Activity

5.7.2.5.1 The total system I/O rate (including network I/O) and the total disk I/O rate, calculated as single averages across the Measurement Interval are required. The average I/O rate, data transfer size and service time for each disk device as seen by the system must also be reported.

5.7.2.5.2 The total I/O rates attributable to database and transaction management logging (for example audit or before image journal) must be reported separately.

5.7.2.6 Web Server Statistics

The following Web Server statistics (some of which may be found in the Web Server Access Log) are required to be reported:

- Connections per second - The number of connections requested by the RBE and accepted by the SUT per second. The intent is to count only the number of new connections made successfully by the RBE in generating the load for the benchmark.

  Comment: Typically, for a sockets based implementation, this statistic would be the number of successful returns from calls to the accept() function. This statistic is not the number of concurrent connections the Web Server handles at any given point in time.

- HTTP requests per second - The number of HTTP requests received per second. The intent is to count only the requests generated by the RBE in generating the load for the benchmark.

Comment: For the purposes of the benchmark, a request is quite likely a GET or a POST request from the RBE. The number of returns from the recv() function may not correctly count the number of requests since it is possible for a request to span multiple calls. In HTTP 1.1, multiple requests may also be sent over one packet (pipelining). The number of requests received is best counted by the parsing code.

- HTTP errors per second - The number of RBE requests that result in an error response per second.

  Comment: This statistic may be obtained by post-processing the Web Server Access Log and counting all 4xx and 5xx response codes. The HTTP specifications defines the following response codes:
  - 1xx Informational
  - 2xx Successful
  - 3xx Redirection
  - 4xx Client Error
  - 5xx Server Error

- Bytes per second - The number of bytes sent to the RBE per second.

  Comment: A server using sockets may obtain this statistic by instrumenting the send() function. The network layer may also be instrumented to provide this statistic.

- Successful returns per second - The number of HTTP 2xx responses codes sent per second.

5.7.3 Optional Performance Monitor Metrics

5.7.3.1 There are many additional data elements which would be of considerable value to a performance analyst but are beyond the scope of the required data described above. In most cases this data is already being reported by the performance monitor being used and would require little additional work. Any optional data reported must be provided by a commercially available performance monitor. Some examples of data which have traditionally been of interest include:

- Breakout of CPU busy by Key, State, Usage, or Software component
- Breakout of true CPU idle and idle wait for I/O
- Queue statistics for work units waiting for the CPU resource
- Page/Swap breakout over specific devices including service times
- Page/Swap breakout by function or data type
- Page/Swap efficiency as pages per I/O
- I/O statistics by device model or performance class
- Utilization and contention for I/O components such as paths, control units, switches, concentrators, IOPs, I/O bus, and interfaces
- Web cache hit rates

5.7.4 Overload Run

5.7.4.1 The intent of the overload run is to simulate the behavior of the SUT under unusually high load. The extra load is simulated by reducing the Think Time to 2 seconds, from the 7 seconds used during the Shopping Interval(see Clause 5.3).

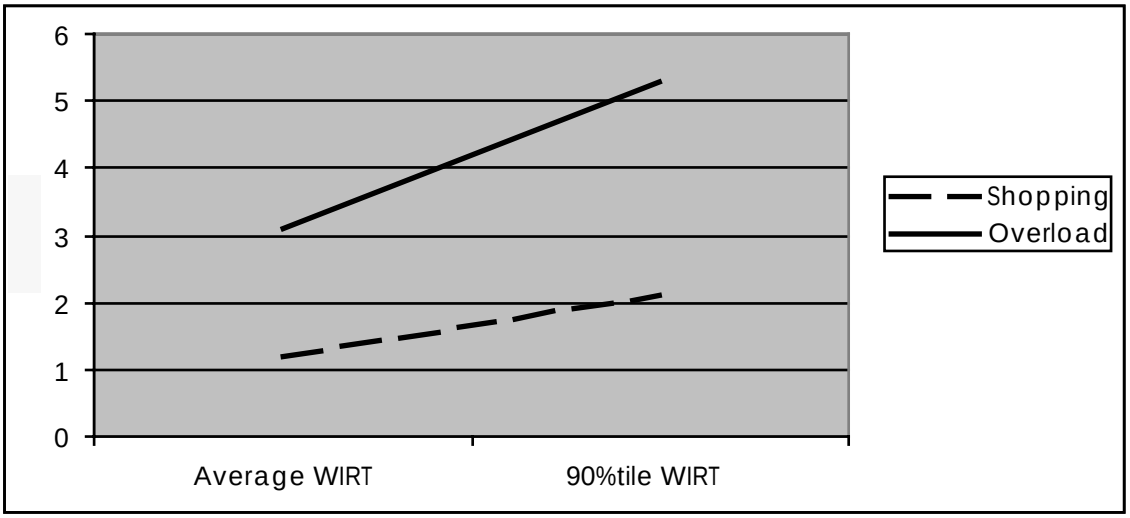
5.7.4.2 The web interaction mix must be the same during the overload run as the one used during the Shopping Interval. The minimum duration for the overload run is 20 minutes after steady state has been reached. WIRT limits and checkpoint requirements do not apply to the overload run.

5.7.4.3 Web interactions may fail under high load, due to Web Server overloading, communication timeouts, or other causes. Every EB must be configured to retry the same interaction whenever a failure occurs. In the case when a retry also fails, the EB should continue to retry as many times as necessary until the interaction succeeds. This is to ensure that the system is capable of surviving high loads without introducing persistent and unrecoverable failures.

5.7.4.4 If a web interaction involves transfer of multiple objects, the objects can be retried individually without re-transferring all objects. However, the web interaction is not considered to be complete until all objects are successfully transferred.

5.7.4.5 The following reports must be included:

- Measured throughput versus elapsed time graph. The graph should conform to the rules in clause 5.6.2, with the only exception that there are fewer intervals because of the shorter duration.
- Home web interaction response time graph. This graph contains four data points: average and 90% WIRT measured in the overloaded run, and the average and 90% WIRT taken for the reported Shopping Measurement Interval. Reporting of WIRT for other web interaction types is not required, due to the volume of data that would be involved. Following is an example of such a graph:



- Web interaction failure rate. A web interaction is considered to have failed if the Web Server responded with an error, or timed-out. The failure rate must be calculated with a precision of 0.01% or higher and must use the following formula:

$$\frac{\text{Total number of retries during the measured interval}}{\text{Total number of web interactions during the measured interval}} \times 100\%$$

If multiple (N) retries occur in a single web interaction, the total number of retries should be incremented by N, not 1.

Clause 6 - SUT, RBE AND NETWORK DEFINITIONS

6.1 Remote Browser Emulator (RBE)

6.1.1 The Remote Browser Emulator (RBE) is the software component that drives the TPC-W workloads. It emulates Users using web browsers to request services from the System Under Test (SUT).

6.1.2 The RBE creates and manages an Emulated Browser (EB) for each emulated User. The term RBE, when used in this specification, includes the entire population of EB.

6.1.3 The RBE communicates with the SUT using TCP/IP sockets.

6.1.4 The RBE is responsible for the following:

- Conforming to all execution rules specified in this document
- Conforming to all industry standards specified in this document, e.g. SSL
- Generating random numbers, timestamps, strings as required to implement the benchmark
- Selecting navigation options for EBs
- Recording the counts and percentages of mix of web interactions requested and completed
- Recording WIRT and throughput in accordance with the requirements of Clause 5
- Conforming to all session timeouts specified in this document

6.1.5 The RBE may not perform any processing ordinarily performed by the SUT. This includes, but is not limited to:

- Caching Web objects, files or data base tables, except as specifically noted in Clause 2, as required to execute the RBE functions.
- Searching files or data bases except as required to execute RBE functions
- Caching disk addresses or pointers to data base records on the SUT
- Composing dynamic web pages or web objects
- Performing computations such as shipping cost, tax, etc. that belong to the SUT
- Contributing to the Promotional Processing specified in Clause 2
- Storing shopping-cart information to avoid processing by the SUT
- Communicating information to the SUT regarding future web interactions or navigation
- Executing active elements (e.g., Java scripts or applets) communicated by the SUT

6.1.6 The RBE may not use proprietary communication protocols when communicating with the SUT. The RBE is required to use TCP/IP sockets (i.e. RFC 1122, etc.) for all network communication. A new socket must be created for eachUser Session. User Sessions cannot share a socket connection. However, the RBE may open multiple socket connections per User Session.

6.1.7 The SUT and RBE must open a new socket connection for each EB. A socket connection cannot be used by more than one EB. Furthermore, if the SUT allows persistent sessions, also known as keep-alive (as specified in HTTP 1.1), the duration parameters for persistent sessions (i.e. time limit and request limit) must be disclosed.

6.1.8 The communication between the EBs and the SUT must not use a TCP Maximum Segment Size (MSS) greater than 1,460 bytes.

6.1.9 User Session Duration

6.1.9.1 The term **User Session Current Duration** is used in this specification to refer to the time, measured by the EB, that has elapsed *from* the time just before the first request by the EB of the Home web interaction *to* the current time during the User Session.

6.1.9.2 The term **User Session Minimum Duration** (**USMD**) is used in this specification to refer to the minimum duration, measured by the EB, that a specific User Session must last. For each User Session, the EB generates a USMD randomly selected from a negative exponential distribution. Each USMD is generated using the following equation:

$$USMD = - \ln(r) * \mu$$

where: ln = natural log (base e)

r = random number uniformly distributed between 0 and 1.0

μ = mean duration of 15 minutes

Comment 1: The resolution of the time stamps must be at least 0.1 seconds.

Comment 2: Each distribution must be truncated at 4 times its mean value (i.e., truncated at 60 minutes, inclusive).

Comment 3: The USMD is defined to support the requirement that a User Session does not end until its USMD has elapsed and the next interaction is going to be a Home interaction (as defined in Clause 6.1.10). Because there will be on average a non-zero time between the USMD elapsing and the next selection of a Home web interaction, the actual average duration for User Sessions will be somewhat greater than μ.

6.1.10 User Session Termination

6.1.10.1 A User Session ends when all of the following conditions hold:

- The EB has just completed the Think Time following the completion of a web interaction.
- The navigation option chosen by the EB for the next web interaction is the Home web interaction.
- The User Session Current Duration is greater than or equal to the User Session Minimum Duration.

6.1.10.2 When the User Session ends, the EB takes the following actions:

- The EB closes any SSL session that is currently established for the User Session.
- The EB closes all network connections that are currently established for the User Session.
- After taking these actions, the EB is allowed to immediately start a new User Session.

6.2 System Under Test (SUT)

The SUT comprises all components which are part of the "application" being simulated. This includes network connections, Web Servers, application servers, database servers, etc.

6.2.1 SUT Contents

The SUT consists of:

- The host system(s) (including hardware and software) required to support the Application and database(s).
- All network components (hardware and software) between host machines which are part of the SUT.
- The network interface to the SUT. Any component of the communications network which use application data within a packet are part of the SUT.

  Comment: The intention is to require any components which are required to form the physical TCP/IP connections (commonly known as the NIC, Network Interface Card) to be part of the SUT. The Internet communications providers are not included in the SUT.
- It does not include the RBE or the PGE.

6.2.2 Functions of the SUT

6.2.2.1 The SUT services the following:

- HTTP GET requests via the HTTP server, either by returning static information from files stored in the file system or by retrieving the information from a database. The database may be accessed using any available interface mechanism;
- HTTP POST requests via the HTTP server, by invoking the appropriate transaction in the application database. The database may be accessed using any available interface mechanism;

6.2.2.2 The SUT performs the following operations:

- All database accesses;
- All communication functions to the EB and PGE;
- All application functionality required to implement the web interactions.

6.2.3 SUT Restrictions

6.2.3.1 The SUT must not perform any caching operations beyond those normally performed by the servers (HTTP, non-HTTP) which are being used.

Comment: The intention is to allow database and Web Server caching to work normally but not to allow the application to take advantage of the limited nature of the benchmark and to cache information which would normally be retrieved from the database.

6.2.3.2 The SUT must not allow any exchanges of information between User Sessions. The context of each User Session must be isolated.

6.2.3.3 No hardware changes can be made to the SUT between the Measurement Intervals of WIPS, WIPSb and WIPSo.

6.3 Payment Gateway Emulator (PGE)

6.3.1 The Payment Gateway Emulator (PGE) represents an external system which authorizes payment of funds as part of purchasing transactions.

6.3.2 The PGE is not included in the SUT.

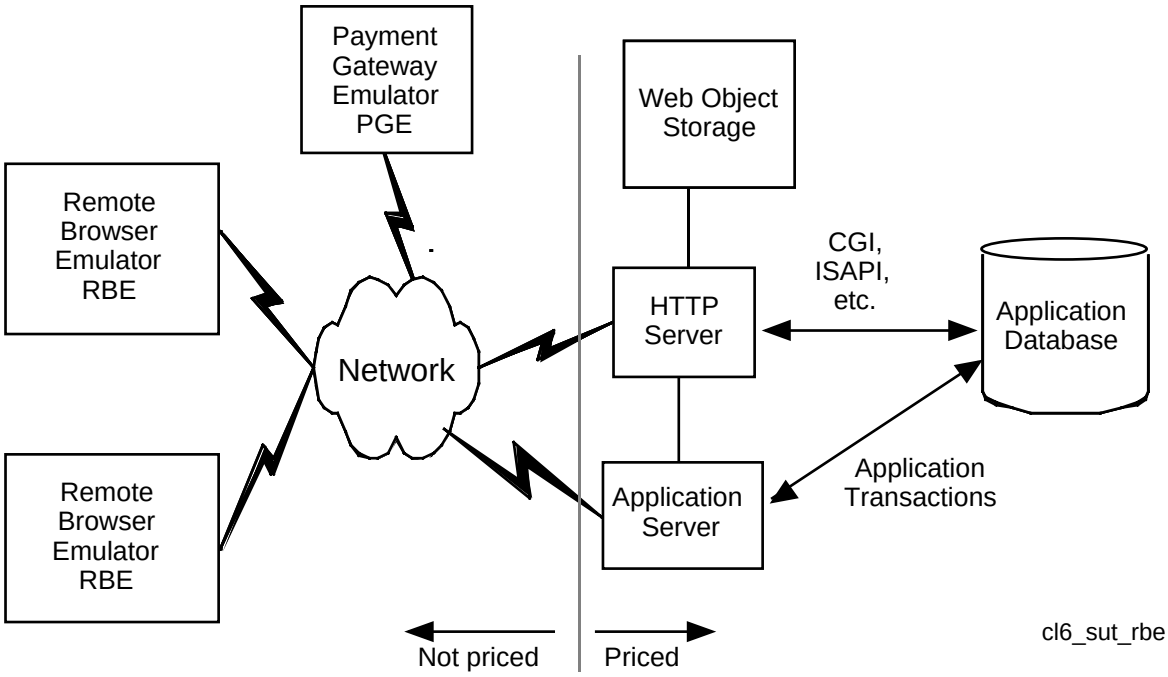
6.3.3 The PGE must perform the following functions:

- Establish an SSL session at the SUT's request
- Receive messages from the SUT
- For each message received from the SUT:
  - Decrypt the messages
  - Extract the O_ID information from the message
  - Generate an authorization code,AUTH_ID, as a random a-string of 15 characters
  - Within the decrypted message at a random location starting after the </OID> tag, embed AUTH_ID preceded by the <CODE> tag and followed by the </CODE> tag.
  - Record the O_ID and authorization number on a durable medium
  - Encrypt the updated response message
  - Send the encrypted response message back to the SUT

6.3.4 The response time between the reception of a given message from the SUT and the communication of the PGE's response message to the SUT must be no less than 2 seconds.

6.4 Model of the Complete Tested System

The following diagram shows an example layout of RBE, PGE and SUT components. All components on the right of the doted line are part of the SUT and, as such, must be priced (see Clause 7).



6.5 Communications Interfaces Definitions

6.5.1 Network Protocol

The network protocol will be TCP/IP.

6.5.2 Application Protocol

The application protocol is the data-level protocol which is used between the RBE and the SUT. All protocols which are used must be commercially available. All web interactions must use the http Protocol, version 1.1 or above. All secure web interactions must use the SSL protocol, version 3 or above.

6.5.3 Communications Within the SUT

The communications protocols within the SUT are not restricted.

6.6 Operational Characteristics of the Merchant Commerce Application

The objective of this benchmark is to represent a 24X7 operating environment for a web-based establishment. The following are requirements of the commerce application that make it consistent with a 24X7 operational model. Some of these functional characteristics may not be actually exercised during benchmark's Measurement Intervals.

6.6.1 Uninterrupted Execution

The commerce application must be capable of continued execution for at least 14 days, where continued execution means 8 hours per day at the reported WIPS throughput and at least 30% of the reported WIPS throughput at other times of the day. The throughput requirements apply within any arbitrary half hour interval (see also Clause 5.5.2.2).

To demonstrate continued execution capability, the following data must be collected for each system that is part of the SUT at the start and end of the reported Shopping Interval (see Clause 5.1.2), prior to any shutdown of the application:

- Total disk space utilization in bytes
- Disk space for system swapping and paging

The total growth in these resources (RG) should be extrapolated to a period of 14 days, assuming an 8-hour day and using the following formula for each resource:

RG = IR + (G / SI * 3600 * 8 * 14)

Where:

SI is the duration of the Shopping Interval in seconds,

G is the growth over the Shopping Interval, and

IR is the initial resource utilization;

The system configuration on which TPC-W benchmarks are run, must be capable of supporting the total resource requirements for 14 days of continuous operation, as defined above. To be considered a valid TPC-W result, the resource requirements for disk, as computed above, must not exceed the capacity of the system.

Comment: If the above requirement is implemented by backing-up some files, such as the Web Server access logs, to tapes at regular intervals, the storage space archived to tape can be excluded from the above storage requirement.

6.6.2 Non-Disruptive Maintenance

The following activities are considered indispensable for normal system maintenance and must be possible during the two weeks of uninterrupted execution, as defined above:

- Site administration
- Security administration
- Offload or archive of system, data base, application or Web Server logs
- Processing accounting information
- Creating Database Checkpoints
- Processing Database Image Copies

Site administration includes the activities of adding or changing html and graphics; adding or changing shopper discounts and advertising messages; and adding or changing special offers.

Security administration involves protecting resources such as data sets, data bases, programs and operator commands from access by unauthorized users. It includes creating generic security profiles, changing permissions for users and resources, allowing the access, update and creation of data sets, and resetting passwords, if necessary.

Processing accounting information refers to the analysis and reporting of accounting data using a program or an accounting software package. Depending upon specific platform implementations, there may also be a requirement to dump or offload accounting records before the analysis program(s) can operate upon them.

Database checkpoints are snapshots of the database at some specific moment in time, for the purposes of establishing a consistent state which can be used for recovery, if necessary. Checkpoints normally occur at predetermined time intervals (e.g. every 25 minutes), when switching from one active log data set to another, after a successful restart or at normal termination.

Database Image Copies are normally created using a database utility program to capture a likeness of the data and/or control information. Image copies may be subsequently used in a recovery operation. Most DBMS support both full image copies (i.e. a dump of the entire database(s)) and incremental image copies (i.e., a copy of the information that has changed since the last image copy).

Clause 7 - PRICING

7.1 Pricing Methodology

7.1.1 The intent of this section is to define the methodology to be used in calculating the 3-year pricing and the price / performance (price / WIPS). The fundamental premise is that what is tested and / or emulated is priced and what is priced is tested and / or emulated. Exceptions to this premise are noted below.

7.1.2 The proposed system to be priced is the aggregation of the SUT components that would be offered to achieve the reported performance level, as defined in Clause 6. Calculation of the priced system consists of:

- Price of the SUT as tested and defined in Clause 6, excluding browsers.
- Price of on-line storage for the database population, 8 hours of processing at the reported WIPS, 14 days of uninterrupted processing and data generated by an additional 180 8-hour days of processing at the reported WIPS.
- The system software necessary to create, operate, administer, and maintain the application program.
- Price of additional products that are required for the operation, administration, performance monitoring or maintenance of the priced system.

7.1.3 The following pricing methodology must be used:

- All hardware and software used in the pricing calculations must be announced and orderable by customers. For any product not already generally released, the Full Disclosure Report must include a committed general delivery date (see Clause 8.1.8). That date must not exceed 6 months beyond the Full Disclosure Report submittal date.

  Comment: Product announcement constitutes public communication about the product in a vehicle other than the FDR, within one calendar week of the TPC-W submission date. The announcement must include a general description of the product. Orderable refers to the ability to obtain the product through the channels promoted as usual and customary for that vendor in selling products of this type.
- Generally available discounts for the priced configuration are permissible.
- Generally available packaged pricing is acceptable.
- Local retail pricing and discount structure should be used in each country for which results are published.
- Price should be represented by the currency with which the customer would purchase the system.
- All hardware components used in the priced system must be new (i.e., not reconditioned or previously owned).
- For test sponsor (s) who have only indirect sales channels, pricing must be actual generally available pricing from indirect channels which meet all other requirements of Clause 7.
- Prices should be shown as whole integer amounts (e.g., dollars but not cents). All fractional amounts should be rounded up to the nearest integer value.

Comment 1: The intent of the pricing methodology is to allow packaging and pricing that is generally available to customers and to explicitly exclude promotional and/or limited availability offerings.

Comment 2: Revenue discounts based on total price are permissible. Any discount must be only for the configuration being priced and cannot be based on past or future purchases. Individually negotiated discounts are not permitted. Special customer discounts (e.g. GSA schedule, educational schedule) are not permitted.

Comment 3: The intent is to benchmark the actual system which the customer would purchase. However, it is realized that vendors may announce new products and disclose benchmark results before the products have actually shipped. This is allowed, but any use of one-of-a-kind hardware/software configurations, which the vendor does not intend to ship in the future is specifically excluded. All products must be generally announced and orderable in the country where the SUT is priced.

7.1.4 The test sponsor (s) must disclose all pricing sources and effective date(s) of the prices.

7.1.5 The sponsor is required to state explicitly all the items and services which are not directly available through the sponsor. Each suppliers items and prices, including discounts, must be listed separately. Discounts may not be dependent on purchases from any other suppliers.

7.1.6 Non-Sponsor Pricing

7.1.6.1 In the event that any hardware, software, or maintenance is provided by a vendor who is not a sponsor of the benchmark, the pricing must satisfy all requirements for general availability, standard volume discounts, and full disclosure. Furthermore, any pricing which is not directly offered by the test sponsor(s) and not derived from the non-sponsoring vendor's generally available pricing and discounts must be guaranteed by the vendor in a written price quotation for a period not less than 60 days from the date the results are submitted. This written quotation must be included in the Full Disclosure Report and state that the quoted prices are generally available, the time period for which the prices are valid, the basis of all discounts, and any terms and conditions which may apply to the quoted prices. The test sponsor must still comply with price changes as described in Clause 8.

7.1.6.2 For items provided by a vendor who is not a sponsor of the benchmark, and for which the aggregated price is less than 1% of the total price, the requirement for a written price quotation from the vendor can be replaced by including in the FDR a copy of the source document from which the price was obtained (e.g., printed advertisement or on-line catalogue page).

7.1.7 Pricing shown in the Full Disclosure Report must reflect line item pricing for hardware, software, and maintenance from the vendor's price books. Line items must reflect quantity one pricing with details providing the basis of any discounts, whether applied to the line item or applied to overall dollar value from a vendor, disclosed separately.

Comment: The intent of this clause is that the pricing reflect the level of detail that an actual customer would see on an itemized billing. The pricing excludes domestic taxes and shipping charges that would be incurred in the country for which the results are published. It is not intended to exclude tariffs, custom duties/fees, and shipping to a domestic port of entry if the component originates in another country.

7.1.8 For publishing in another country other than the country for which the results are originally published, it is permitted to substitute local components from the original report providing the substituted products are sold to the same product description or specifications.

Comment: The intent of this clause is to encourage local country pricing by allowing substitution of equipment for country specific reasons such as voltage, product numbering, industrial/safety, keyboard differences, etc., which do not affect performance.

7.2 Priced System

7.2.1 SUT

The entire price of the SUT as configured during the test must be used, including all hardware (new purchase price), software (license charges) and hardware/software maintenance charges over a period of 3 years (36 months). In the case where the Driver System provides functionality in addition to the RBE described in Clause 6, then the price of the emulated hardware/software described in Clause 7.2.2.1 are to be included.

Comment 1: The intent is to price the tested system at the full price a customer would pay. Specifically prohibited are the assumption of other purchases, other sites with similar systems, or any other assumption which relies on the principle that the customer has made any other purchase from the vendor. This is a one time, stand-alone purchase.

Comment 2: The number of users for TPC-W is defined to be equal to the number of browsers emulated in the tested configuration. Any usage pricing for the above number of users should be based on the pricing policy of the company supplying the priced component.

7.2.2 Browsers and Network Pricing

7.2.2.1 The price of the Remote Browser Emulator (RBE) and the price of the SET Payment Gateway Emulator (PGE) are not included in the pricing calculation. Please refer to Clause 6.2.1 for a description of the components of the SUT, all of which must be priced.

7.2.3 Database Storage and Recovery Log Pricing

7.2.3.1 Within the priced system, there must be sufficient on-line storage to support any expanding system files and the durable database population resulting from executing the TPC-W transaction mix for 180 eight hour days at the reported WIPS (see Clause 4.2.3). Storage is considered on-line, if any record can be accessed randomly and updated within 1 second. On-line storage may include magnetic disks, optical disks, or any combination of these, provided that the above mentioned access criteria is met.

Comment 1: The intent of this clause is to consider as on-line any storage device capable of providing an access time to data, for random read or update, of one second or less, even if this access time requires the creation of a logical access path not present in the tested database. For example, a disk based sequential file might require the creation of an index to satisfy the access time requirement.

7.2.3.2 Recovery data must be maintained in such a way that the published WIPS rate could be sustained for an 8-hour period. Roll-back recovery data must be either in memory or in on-line storage at least until transactions are committed. Roll-forward recovery data may be stored on-line. All ACID properties must be maintained.

7.2.3.3 It is permissible to not have the storage required for the 180-day space on the tested system. However, any additional storage device included in the priced system but not configured on the tested system must be of the type(s) actually used during the test and must satisfy normal system configuration rules.

Comment: Storage devices are considered to be of the same type if they are identical in all aspects of their product description and technical specifications.

7.2.3.4 The requirement to support eight hours of recovery log data can be met with storage on any durable media (see Clause 3.5.1) if all data required for recovery from failures listed in Clauses 3.5.3.2 and 3.5.3.3 are on-line.

7.2.4 Additional Operational Components

7.2.4.1 Additional products that might be included on a customer installed configuration, such as operator consoles and magnetic tape drives, are also to be included in the priced system if explicitly required for the operation, administration, or maintenance, of the priced system.

7.2.4.2 Copies of the software, on appropriate media, and a software load device, if required for initial load or maintenance updates, must be included.

7.2.4.3 The price of an Uninterruptible Power Supply, specifically contributing to a durability solution, must be included (see Clause 3.5.1).

7.2.4.4 The price of all cables used to connect components of the system (except as noted in Clause 7.2.2.3) must be included.

7.2.5 Additional Software

7.2.5.1 The price must include the software licenses necessary to create, compile, link, and execute this benchmark application as well as all run-time licenses required to execute on host system(s), client system(s) and connected workstation(s) if used.

7.2.5.2 In the event the application program is developed on a system other than the SUT, the price of that system and any compilers and other software used must also be included as part of the priced system.

7.3 Maintenance

7.3.1.1 Hardware and software maintenance must be figured at a standard pricing which covers at least 7 days/week, 24 hours/day coverage, either on-site, or if available as standard offering, via a central support facility. Hardware maintenance maximum response time must not exceed 4 hours, on any part whose replacement is necessary for the resumption of operation. The 4-hour maintenance response time requirement may be met by customer spareable and replaceable hardware, as described in Clause 7.3.5.

Comment 1: Resumption of operation means the priced system must be returned to the same configuration that was present before the failure.

Comment 2: The intent of hardware maintenance pricing is not met by pricing based on the cost to fix specific failures, even if the failure rate is calculated from Mean Time Between Failure (MTBF). The maintenance pricing must be independent of actual failure rates over the 3 year period, no matter how many failures occur during that period. The intent is to preclude the use of MTBF to directly compute the maintenance cost for this benchmark.

7.3.2 If central support is claimed, then the appropriate connection device, such as auto-dial modem must be included in the hardware price. Also any software required to run the connection to the central support, as well as any diagnostic software which the central support facility requires to be resident on the tested system, must not only be included in pricing, but must also be installed during the benchmark runs.

7.3.3 Software maintenance must include maintenance update distribution for both the software and its documentation. If software maintenance updates are separately priced, then pricing must include at least 2 updates over the 3 year period.

Comment: Software maintenance, as defined above, means a standard offering which includes acknowledgment of new and existing problems within 4 hours and a commitment to fix defects within a reasonable time.

7.3.4 It is acceptable to incorporate, for pricing purposes, the use of customer spareable and replaceable hardware items under the following conditions:

- An additional 10% of the number of configured units of the replaceable items, with a minimum of 2, must be priced for spares.
- The vendor must include a support service which guaranties replenishment on-site within 7 days throughout the 3 year maintenance period and covers every unit of the replaceable items (i.e., configured units and spare units).
- The items must be generally available as spareable and replaceable for any customer installation.
- The designation of the items as spareable and replaceable cannot depend on a threshold of purchased quantity.
- It must be verifiable that the action of diagnosing the spareable and replaceable items as having failed can be positively accomplished by the customer within 4 hours of failure.
- The method for diagnosis and replacement of the replaceable items must have complete customer documentation.

Comment 1: Diagnosis may take the form of a hardware indicator or a diagnosis procedure. The intent is that the diagnosis must reach a positive conclusion as to the state of the item within 4 hours.

Comment 2: The use of spares is intended to assist in complying with the 4-hour maximum hardware maintenance response requirement. It cannot be a substitute for maintenance support, as the priced configuration must maintain the same quantities of components, including spares, for 3 years.

7.4 Required Reporting

7.4.1 Two metrics will be reported with regard to pricing. The first is the total 3 year pricing as described in the previous clauses. The second is the total 3 year pricing divided by the reported Web Interactions Per Second (WIPS), as defined in Clause 5.4.

7.4.2 The 3-year pricing metric must be fully reported in the basic monetary unit of the local currency (see Clause 7.1.3) rounded up and the price / performance metric must be reported to a minimum precision of three significant digits rounded up. Neither metric may be interpolated or extrapolated. For example, if the total price is US$ 5,734,417.89 and the reported throughput is 105 WIPS, then the 3 year pricing is US$ 5,734,418 and the price / performance is US$ 54,700 / WIPS (5,734,418 / 105).

Clause 8 - FULL DISCLOSURE REPORT

8.1 General Requirements

8.1.1 A Full Disclosure report is required in order for results to be considered compliant with the TPC-W benchmark specification.

Comment: The intent of this disclosure is for a customer to be able to replicate the results of this benchmark given the appropriate documentation and products. This section includes a list of requirements for the Full Disclosure report.

8.1.2 The order and titles of sections in the Test Sponsor's Full Disclosure report must correspond with the order and titles of sections from the TPC-W standard specification (i.e., this document). The intent is to make it as easy as possible for readers to compare and contrast material in different Full Disclosure reports. All sections of the report, including appendices, must be printed using a minimum font size of 10 points.

8.1.3 The TPC Executive Summary Statement must be included as the first pages of the Full Disclosure report. An example of the Executive Summary Statement is presented in Appendix H. The latest version of the required format is available from the TPC Administrator.

8.1.4 A statement identifying the benchmark sponsor(s) and other participating companies must be provided.

8.1.5 The numerical quantities listed below must be included in the Executive Summary in a tablular form giving the following quantities for each of the reported WIPS, WIPSb and WIPSo measurement intervals:

- The scale factor (cardinality of the ITEM table)
- The measurement interval in minutes
- The WIPS@scale factor, WIPSb@scale factor, and WIPSo@scale factor measured during the intervals
- The measured interaction mix for each of the intervals
- Ninetieth percentile, average and maximum response times for each interaction type

Comment: Appendix H contains an example of such a summary. The intent is for data to be conveniently and easily accessible in a familiar arrangement and style. It is not required to precisely mimic the layout shown in Appendix H.

8.1.6 The application program or programs as defined in 1.2.8 must be disclosed.

8.1.7 Settings must be provided for all customer-tunable parameters and options which have been changed from the defaults found in actual products, including but not limited to:

- Database tuning parameters.
- Web server tuning and logging options.
- Recovery / commit options.

- Consistency / locking options.
- Operating system and application configuration parameters.
- Compilation and linkage options and run-time optimizations used to create / install applications, OS, DBMS, web server, and / or any other commercial product.

Comment 1: This requirement can be satisfied by providing a full list of all parameters and options.

Comment 2: The intent of the above clause is that anyone attempting to recreate the benchmark environment has sufficient information to compile, link, optimize, and execute all software used to produce the disclosed benchmark result.

8.1.8 Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

- Number and type of processors.
- Physical memory actually present on the SUT.
- Size of allocated memory, and any specific mapping / partitioning of memory unique to the test.
- Number and type of disk units and controllers.
- Number of channels or bus connections to disk units, including their protocol type.
- Number of LAN (e.g., Ethernet) connections, including routers, workstations, terminals, etc., that were physically used in the test or are incorporated into the pricing structure.
- Type and the run-time execution location of software components (e.g., DBMS, web server, application server or progam, transaction monitors, etc.).

Comment: Detailed diagrams for system configurations and architectures can widely vary, and it is impossible to provide exact guidelines suitable for all implementations. The intent here is to describe the system components and connections in sufficient detail to allow independent reconstruction of the measurement environment.

8.2 Clause 1 - Web Object and Logical Database Design

8.2.1 The location of the JPEG and GIF images used in the benchmark must be disclosed . This information must include on which system and in what logical structure the images are stored (database or file system).

8.2.2 Listings must be provided for all table definition statements and all other statements used to set-up the database.

8.2.3 The physical organization of tables and indices, within the database, must be disclosed.

Comment: The concept of physical organization includes, but is not limited to: record clustering (i.e., rows from different logical tables are co-located on the same physical data page), index clustering (i.e., rows and leaf nodes of an index to these rows are co-located on the same physical data page), and partial fill-factors (i.e., physical data pages are left partially empty even though additional rows are available to fill them).

8.2.4 Any horizontal or vertical partitioning of tables or rows in the TPC-W benchmark must be disclosed (see Clause 1.6.4 and 1.6.5). Replication of tables, if used, must be disclosed (see Clause 1.6.6).

8.2.5 Additional and/or duplicated attributes in any table must be disclosed along with a statement on the impact on performance (see Clause 1.6.7).

8.3 Clause 2 - Web Interactions and Workload

8.3.1 The method of random number generation must be disclosed.

8.3.2 The number of Emulated Browsers used in the measured intervals must be disclosed.

8.3.3 The percentage of User sessions started during each measurement interval with known Customers must be disclosed with at least 1 decimal place.

8.3.4 The implementation of the CART data must be explained in sufficient detail to allow a replication of the benchmark including the durability of the CART data over a single point of failure.

8.3.5 A description of how the security requirements were met as defined in Clause 2.2.1, must be disclosed including key lengths.

8.3.6 The HTML code for one representative web page for each web interaction must be included in the Full Disclosure Report.

8.3.7 The mix (i.e., percentages) of transaction types seen by the SUT during each measured interval must be disclosed.

8.3.8 A statement must be provided describing the development language(s) and the types of API's used between commercial components to implement the interactions. This includes, but is not limited to, the interfaces to the database server, web server, commerce package or application, or any other commericial product used. For example: The commerce package calls application code through standard CGI interface. The application is written in C and interfaces to the database with ODBC calls that initiate stored procedures.

8.4 Clause 3 - Database Transaction and System Properties

8.4.1 The results of the ACID tests must be disclosed along with a description of how the ACID requirements were met.

8.4.2 The method for meeting Web Page consistency requirements (Clause 3.2) must be described.

8.5 Clause 4 - Scaling and Database Population

8.5.1 The cardinality (e.g., number of rows) of each table, as initially populated (see Clause 4.3 & 4.4), must be disclosed.

8.5.2 The space required to sustain 180 days of the reported throughput as defined in Clause 4.4 must be disclosed.

8.5.3 The space required for 24 hours of Web-Server access logs as defined in Clause 4.5 must be disclosed.

8.5.4 The method for distributing table and log data across all media must be described. A detailed diagram or listing of database files indicating the disks or volumes on which they reside must be included. Simple diagrams can be used for clarification.

Comment: Detailed diagrams for layout of database files on disks can widely vary, and it is difficult to provide exact guideline suitable for all implementations. The intent is to provide sufficient detail to allow independent reconstruction of the test database and access logs.

8.6 Clause 5 Performance Metrics and Response Times

8.6.1 The WIPS@scale factor and price per WIPS@scale factor must be disclosed. WIPSb@scale factor and WIPSo@scale factor are secondary metrics and must be included in the Executive Summary.

8.6.2 A description of the method used to determine the reproducibility of the measurement results must be reported as well as the WIPS from the reporoducibility run.

8.6.3 The duration, start time and stop time of each measurement interval must be disclosed. For measurement intervals reported from the same run, the timing precision must be sufficiently small to demonstrate each measurement interval is non-overlapping.

8.6.4 The percentage of each web interaction executed during each measured interval must be displayed in tabular form as in the following example. Percentages must be reported to 2 decimal points.

Web Interaction	Browsing Mix (WIPSb)	Shopping Mix (WIPS)	OLTP Mix (WIPSo)
Browse	**95.00 %**	**80.00 %**	**50.00 %**
Home	29.00 %	16.00 %	9.12 %
New Product	11.00%	5.00%	0.46%
Best Seller	11.00 %	5.00 %	0.46 %
Product Detail	21.00 %	17.00 %	12.35 %
Search Request	12.00 %	20.00 %	14.54 %
Search Results	11.00%	17.00%	13.08%
Order	**5.00 %**	**20.00 %**	**50.00 %**
Shopping Cart	2.00 %	11.60 %	13.53 %
Customer Reg.	0.82%	3.00%	12.86%
Buy Request	0.75%	2.60%	12.73%
Buy Confirm	0.69%	1.20%	10.18%
Order Inquiry	0.30%	0.75%	0.25%
Order Display	0.25 %	0.66%	0.22%
Admin Request	0.10%	0.10%	0.12%
Admin Confirm	0.09%	0.09%	0.11%

8.6.5 The minimum, maximum, average and 90th percentile response time must be reported for each web interaction.

8.6.6 The minimum, maximum and average think time must be reported for each web interaction.

8.6.7 The number of checkpoints in the Measurement Interval, the time in seconds from the start of the Measurement Interval to the first checkpoint and the Checkpoint Interval must be disclosed.

8.6.8 The frequency distribution of all response times of all web interactions must be graphed as described in Clause 5.6.1 and the graphs must be included in the Full Disclosure Report.

8.6.9 The measured throughput of each measured interval must be graphed as described in Clause 5.6.2 and the graphs be included in the Full Disclosure Report.

8.6.10 A set of monitored metrics is required and must be reported. Clause 5.7.1 and 5.7.2 define how these monitored metrics are to be aggregated.

8.6.11 The monitoring tools used during the measurement interval must be described and the operational methods invoked must be reported. For example, the monitor was invoked as a started task, and binary data was recorded every 15 seconds to a disk for post-processing.

8.6.12 If a sampling technique is used to obtain performance data then the sample rate (per second) must be reported as specified in Clause 5.7.1.6

8.6.13 The monitored metrics as defined in Clause 5.7 must be reported in a table similar to the following example:

Metric	HTTP Server Aggregates	DB Server Aggregates
% CPU busy		
Physical memory		
Memory dedic. to shared pool		
Page size of memory manager		
Page In / Page Out		
Swap In / Swap Out		
Logical READS per second		
Physical READS per second		
Logical WRITES per second		
Physical WRITES per second		
System I/O rate		
Total Disk I/O rate		
Avg device utilization		
Avg service time		
Avg disk data transfer size		
Total logging I/O rates		
TCP/IP connections / second		
HTTP requests / sec		
HTTP errors / sec		
HTTP bytes / sec		
HTTP successes / sec		

8.6.14 The CPU utilization must be graphed (as defined in clause 5.7.2.1.2) and the graph must be included in the Full Disclosure Report.

8.7 Clause 6 - SUT, RBE and Network

8.7.1 A complete functional diagram of the measured configuration must be disclosed. A detailed list of all software and hardware functionality being performed on the Driver System, and its interface to the SUT must be disclosed (see Clause 6.6.3.6).

8.7.2 The rated bandwidth of the network(s) component used in the measured configuration must be disclosed along with any setting restricitng that bandwidth. This includes, but is not limited to, the internode connections withing SUT, between the SUT and the RBE, and between the SUT and the Payment Gateway Emulator.

8.7.3 If the configuration requires operator intervention to meet the requirements of performance levels and uninterrupted operations, the mechanism and the frequency of this intervention must be disclosed.

8.7.4 It must be disclosed if the RBE is commercially available or proprietary.

8.8 Clause 7 - Pricing

8.8.1 A detailed list of hardware and software used in the priced system must be reported in the Executive Summary. Each separately orderable item must have vendor part number, description, and release/revision level, and either general availability status or committed delivery date. If package-pricing is used, vendor part number of the package and a description uniquely identifying each of the components of the package must be disclosed. Pricing source(s) and effective date(s) of price(s) must also be reported. An example of the Executive Summary Statement is presented in Appendix H.

8.8.2 The total 3-year price of the entire configuration must be reported, including: hardware, software, and maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.

8.8.3 The committed delivery date for general availability (availability date) of products used in the price calculations must be reported. When the priced system includes products with different availability dates, the reported availability date for the priced system must be the date at which all components are committed to be available. This single date must be reported on the first page of the Executive Summary. All availability dates, whether for individual components or for the SUT as a whole, must be disclosed to a precision of one day.

8.8.4 For any usage pricing, the sponsor must disclose:

- Usage level at which the component was priced.
- A statement of the company policy allowing such pricing.

Comment: Usage pricing may include, but is not limited to, the operating system and database management software. System pricing must include line item indication where non-sponsoring companies' brands are used. System pricing must also include line item indication of third party pricing.

8.9 Clause 9 - Audit Related Items

8.9.1 The auditor's name, address, phone number, and a copy of the auditor's attestation letter indicating compliance must be included in the Full Disclosure Report.

8.9.2 A review of the pricing model is required to ensure that all components required are priced (see Clause 9.2.8). The auditor is not required to review the final Full Disclosure Report or the final pricing prior to issuing the attestation letter.

8.10 Availability of the Full Disclosure Report

8.10.1 The Full Disclosure Report must be readily available to the public at a reasonable charge, similar to charges for similar documents by that test sponsor.

8.10.2 The report must be made available when results are made public.

8.10.3 The official full disclosure report must be available in English but may be translated to additional languages.

8.11 Revisions to the Full Disclosure Report

8.11.1 Required Revisions

Revisions to the full disclosure documentation are **required** to be published under the following circumstances:

8.11.1.1 During the normal product life cycle problems will be uncovered that require changes, sometimes referred to as ECOs, FCOs, Patches, Updates, etc. When the cumulative result of applied changes causes the WIPS rating of the system to decrease by more than 2% from the reported WIPS, then the test sponsor is required to re-validate the benchmark results and publish the revised performance and price/performance.

8.11.1.2 When cumulative price changes have resulted in an increase of 2% or more from the reported price/performance, the test sponsor must submit revised price/performance results to the TPC within 30 days of the effective date of the price change(s) to remain in compliance. The benchmark need not be re-run to remain compliant.

Comment: The intent of this clause is that published price/performance reflect actual current price/performance.

8.11.1.3 Re-pricing of current results must be reviewed and approved by the auditor if there is a change to the pricing model. Changes in prices of line item components do not constitute a pricing model change.

8.11.1.4 A change in the committed availability date for the priced system which is later than the published availability date must published in a revised Full Disclosure Report.

8.11.1.5 Hardware or Software product substitutions within the SUT, with the exceptions noted below require the benchmark to be re-run with the new components in order to re-establish compliance. The exceptions are:

- Secondary components such as disks, terminal servers, network adapters, routers, bridges, repeaters and the like may be substituted. The substitute must be demonstrated to be at least equivalent to the original in performance. The performance tests and data used to demonstrate equal or greater performance must be disclosed.
- In addition, the capacity of the substitute (memory, slots for systems, formatted MB for disks, etc.) must be at least equivalent to the original.

Comment 1: The intent is to allow substitutions when the change would produce performance at least equivalent to the reported WIPS. The auditor may require additional tests to be run if the proof by documentation is not considered adequate. The auditor's letter of attestation must be attached to the revised full disclosure report.

Comment 2: Substitution of any primary component of the HTTP server or DB server is not allowed. This includes, but is not limited to, CPUs, controllers, internode connectors, channel adapters or memory chips. New versions of the OS, DBMS, or TP Monitor are not allowed under any circumstances.

Comment 3: The component substitution will be open to challenge for a 60 day period.

8.11.1.6 Full Disclosure Report revisions may be required for other reasons according to TPC policies (see TPC Policy Document).

8.11.1.7 The revised report should be submitted as defined in Clause 8.10.

8.11.2 Permitted Revisions

Revisions to the full disclosure documentation are **permitted** to be published under the following circumstances:

8.11.2.1 When cumulative price changes have resulted in a decrease of 2% or more from the reported price/performance, the test sponsor may submit revised price/performance results to the TPC. The benchmark need not be re-run to remain compliant.

8.11.2.2 Re-pricing of current results must be reviewed and approved by the auditor if there is a change to the pricing model. Changes in prices of line item components do not constitute a pricing model change.

8.11.2.3 A change in the committed availability date for the priced system which is earlier than the published availability date may published in a revised Full Disclosure Report.

8.11.2.4 A report may be revised to add or delete Clause 7 related items for country specific priced configurations.

8.11.2.5 The revised report should be submitted as defined in Clause 8.10.

Clause 9 - AUDIT

9.1 General Rules

9.1.1 An independent audit of the benchmark results by an auditor certified by the TPC is required. An audit checklist is provided as part of this specification. Please obtain the current audit checklist from one of the auditors. The term "independent" is defined as: "the outcome of the benchmark carries no financial benefit to the auditing agency other than fees earned directly related to the audit." In addition, the auditing agency cannot have supplied any performance consulting under contract for the benchmark under audit. The term "certified" is defined as: "the TPC has reviewed the qualification of the auditor and certified that the auditor is capable of verifying compliance of the benchmark result." Please see the TPC Audit Policy for a detailed description of the auditor certification process.

In addition, the following conditions must be met:

1. The auditing agency cannot be financially related to the sponsor. For example, the auditing agency is financially related if it is a dependent division, the majority of its stock is owned by the sponsor, etc.
2. The auditing agency cannot be financially related to any one of the suppliers of the measured or priced components, This includes but is not limited to the DBMS supplier, the web server supplier, or any other hardware or software supplier used in the system under test.

9.1.2 The auditor's attestation letter must be made readily available to the public as part of the Full Disclosure Report, but a detailed report from the auditor is not required.

9.1.3 In the case of audited TPC-W results which are being revised for pricing or substitution changes, the audit requirements are called out in Clause 8.11. The auditor is not required to follow any of the remaining auditor's check list items from Clause 9.2.

9.2 Auditor's Check List

9.2.1 Clause 0 – Commercial products

9.2.1.1 Verify that the products measured and priced are orderable by customer as defined in Clause 7.1.3.

9.2.2 Clause 1 – Web Object and Logical Database Design

9.2.2.1 Verify that specified minimum structure (i.e., list of fields) exist for each table, and that they conform to the specified requirements.

9.2.2.2 Verify the web objects are generated according to Clause 1.5.

9.2.2.3 Verify that the unique identifiers in each table are not disk or file offsets.

9.2.2.4 Verify that the storage of the web objects complies with the implementation rules.

9.2.2.5 Verify that all tables support retrievals, inserts and deletes while supporting a TPC-W workload.

Comment: For this verification, the workload does not have to be equal to the reported performance.

9.2.2.6 Verify whether any horizontal and/or vertical partitioning has been used, and, if so, whether the details of that partitioning are disclosed.

9.2.2.7 Verify whether any replication of tables has been used, and, if so, whether at least one copy is durable and if the details of that replication are disclosed.

9.2.2.8 Verify that the integrity rules defined in Clause 1.7 are followed.

9.2.2.9 Verify that the requirements for transparent data access, as defined in Clause 1.8, are met.

9.2.3 Clause 2 – Web Interactions and Workload Profile

9.2.3.1 Verify that the secure web interactions use encrypted communication as defined in Clause 2.2.1.

9.2.3.2 Verify that User Sessions are correctly started as defined in Clause 2.2.3.

9.2.3.3 Verify that the communication of the Customer's identity to the SUT complies with the requirements of Clause 2.2.4 and that Session tracking is implemented according to Clause 2.2.5.

9.2.3.4 Verify that the implementation of the Shopping Cart meets the requirements of Clause 2.2.6.

9.2.3.5 Verify that web logging is turned on for Common Log Format (CLF) from the http server, and meets the requirements defined in Clause 2.2.13.

9.2.3.6 Verify that the application program correctly implements the generic Promotional Processing.

9.2.3.7 Verify the compliant implementation of the Home web interaction.

9.2.3.8 Verify the compliant implementation of the Shopping Cart web interaction.

9.2.3.9 Verify the compliant implementation of the Customer Registration web interaction.

9.2.3.10 Verify the compliant implementation of the Buy Request web interaction.

9.2.3.11 Verify the compliant implementation of the Buy Confirm web interaction.

9.2.3.12 Verify the compliant implementation of the Order Inquiry web interaction.

9.2.3.13 Verify the compliant implementation of the Order Display web interaction.

9.2.3.14 Verify the compliant implementation of the Search Request web interaction.

9.2.3.15 Verify the compliant implementation of the Search Result web interaction.

9.2.3.16 Verify the compliant implementation of the New Products web interaction.

9.2.3.17 Verify the compliant implementation of the Best Sellers web interaction.

9.2.3.18 Verify the compliant implementation of the Product Detail web interaction.

9.2.3.19 Verify the compliant implementation of the Admin Request web interaction.

9.2.3.20 Verify the compliant implementation of the Admin Confirm web interaction.

9.2.3.21 Using one of the configured browsers, verify that the input/output screen for each type of web interaction has the required look and feel and provides all the required features, such as data entry and navigation.

9.2.4 Clause 3 - Database Transaction and System Properties

9.2.4.1 Verify that the required atomicity property is provided by the implementation of the benchmark.

9.2.4.2 Verify the proper execution of the atomicity test, and define and verify the execution of additional tests if necessary to sufficiently demonstrate the atomicity property.

9.2.4.3 Verify that the required consistency property is provided by the implementation of the benchmark.

9.2.4.4 Verify the proper execution of the consistency test, and define and verify the execution of additional tests if necessary to sufficiently demonstrate the consistency property.

9.2.4.5 Verify that the required isolation property is provided by the implementation of the benchmark.

9.2.4.6 Verify the proper execution of the isolation tests, and define and verify the execution of additional tests if necessary to sufficiently demonstrate the isolation property.

9.2.4.7 Verify that the required durability property is provided by the implementation of the benchmark.

9.2.4.8 Determine which component of the SUT meet the definition of a single point of failure as defined in Clause 3.1.5.3.

9.2.4.9 Verify the proper execution of the durability tests involving each single point of failure, and define and verify the execution of additional tests if necessary to sufficiently demonstrate the durability property.

9.2.4.10 Verify that the required web page consistency property is provided by the implementation of the benchmark.

9.2.4.11 Verify the proper execution of the web page consistency test, and define and verify the execution of additional tests if necessary to sufficiently demonstrate the consistency property.

9.2.5 Clause 4 - Scaling and Database Population

9.2.5.1 Verify that the initial database population is properly scaled, as defined in Clause 4.2.

9.2.5.2 Verify that the database population was generated according to Clauses 4.6 and 4.7.

9.2.5.3 Verify the correct cardinalities of the database tables, at the start of the Test Run as well as at the end of it, and that the growth in the ORDERS table, in particular, is consistent with the reported WIPS rate.

9.2.5.4 Verify that the 180-day space requirement is computed according to Clause 4.4.

9.2.5.5 Verify that the size of the web server access logs is calculated according to the requirements.

9.2.5.6 Verify that the required number of EB's are active during the entire Test Run.

9.2.6 Clause 5 - Performance Metrics and Response Time

9.2.6.1 Verify that the mix of web interactions as seen by the RBE satisfies the required mix for each Measurement Interval.

9.2.6.2 Verify that the thresholds used to achieve the required mix of web interactions are those given in Clause 5.2.2.

9.2.6.3 Verify that the pacing of web interactions by the RBE meets the requirements illustrated in Clause 5.2.3.

9.2.6.4 Verify that the Think Times are properly generated, measured and reported by the RBE.

9.2.6.5 Verify the duration of the reported measurement intervals.

9.2.6.6 Verify the validity of the method used to accurately measure and report the Web Interaction Response Times (WIRT) at the RBE.

9.2.6.7 Verify that all measured WIRT's meet the response time constraints defined in Clause 5.3.4.1.

9.2.6.8 Verify that the reported WIRT's are measured during the same time interval as the Measurement Interval from which the reported throughput is measured.

9.2.6.9 Verify that the throughput ratings for the three Measurement Intervals are computed according to Clause 5.4.

9.2.6.10 Verify that the requirements for the Test Run(s), as specified in Clause 5.5.2, are met.

9.2.6.11 Verify the method used to determine that the SUT had reached steady state prior to commencing the Measurement Intervals, as specified in Clause 5.5.2.

9.2.6.12 Verify that the SUT is capable of running uninterrupted for 14 days, as defined in Clause 5.5.2.2. While an actual 14 day test run would be highly impractical, the auditor must conclude with a high degree of confidence that this requirement is met by the SUT. This can be achieved through theoretical explanations and, if needed, the definition and execution of additional tests.

9.2.6.13 Verify that all work normally done in a steady state environment, such as checkpoints or redo/undo logging, actually occurred during the Measurement Intervals in order to keep throughput variations over an 8 hour period under 5%.

9.2.6.14 Verify the method used to determine the reproducibility of the measurement results.

9.2.6.15 Verify that all required reporting metrics, including the additional performance metrics defined in Clause 5.7 are collected accurately and during the same time interval as the Measurement Interval from which the reported throughput is measured.

9.2.6.16 Verify that the overload run is executed according to the requirements of Clause 5.7.4 and that the required report is based on the data measured during that run.

9.2.7 Clause 6 - SUT, Driver, and Communications Definition

9.2.7.1 Verify the accurate emulation of the tested browser population by the RBE.

9.2.7.2 Verify that the RBE correctly implements the CURL, as defined in Clause 2.2.14.

9.2.7.3 Verify that the restrictions placed on the activity of the RBE, as defined in Clause 6.1.5, are observed.

9.2.7.4 Verify that the required communication protocol is used between the RBE and the SUT.

9.2.7.5 Verify that each EB opens its own socket connection to the SUT instead of sharing a socket with other EBs.

9.2.7.6 Verify that the TCP Maximum Segment Size (MSS) between the EBs and the SUT is limited as defined in Clause 6.1.7.

9.2.7.7 Verify that the EBs generate and observe USMDs as defined in Clause 6.1.9.2.

9.2.7.8 Verify that the SUT maintains User context as required.

9.2.7.9 Verify that the restrictions placed on the activity of the SUT are observed.

9.2.7.10 Verify the SUT meets the operational requirements defined in Clause 6.6.

9.2.7.11 Verify the accurate implementation of the PGE functions.

9.2.8 Clause 7 - Pricing

9.2.8.1 Verify that the pricing model includes all the hardware and software components required to produce a compliant benchmark result, including but not limited to sufficient database and log storage.

9.2.8.2 Verify the pricing spreadsheet that details how the 3-year cost of ownership is computed.

9.2.8.3 Verify that, when used, customer spareable components are priced in sufficient quantity to meet the requirements.

9.2.8.4 Verify the compliance of any discount, warranty or maintenance used to meet the pricing requirements.

9.2.8.5 Verify the compliance of all price quotes for components from third party sources.

9.2.8.6 The auditor is not required to verify the pricing of components supplied by the benchmark sponsor.

9.2.8.7 Verify that the price performance ratio is computed according to Clause 7.4.

9.2.9 Clause 8 - FDR

9.2.9.1 Verify that the Full Disclosure Report is accurate. Auditors can limit their review of the FDR to the following sections:

- Executive summary section

- System configuration section
- Database population and space calculation section
- Sections that include the disclosure of measurement data
- Any other section requested by the auditor

Appendix A – State Diagram of C_ID

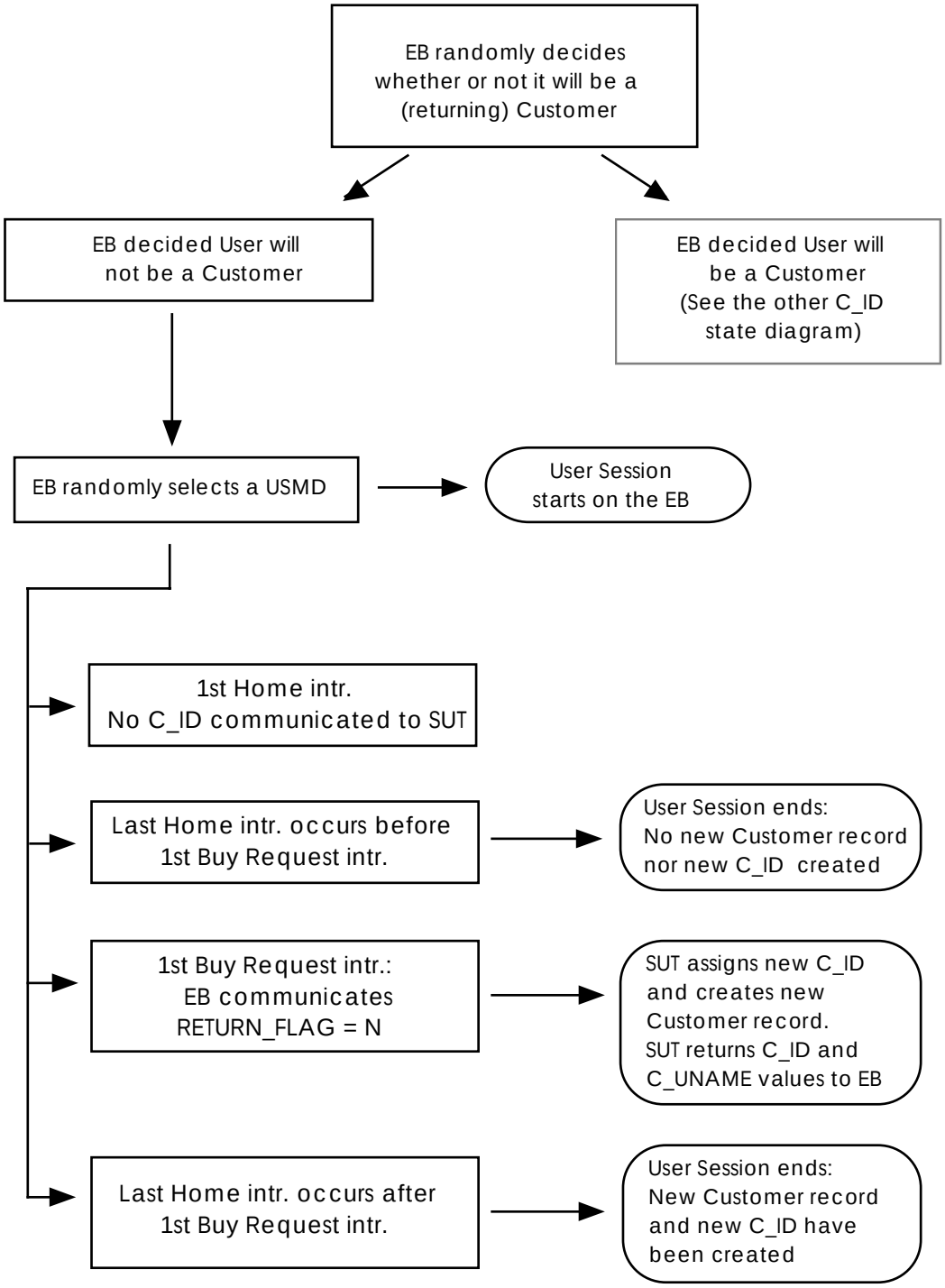
A.1 Introductory Explanations

The two state transition diagrams that follow illustrate the dynamic between the RBE (and its EBs) and the SUT as it pertains to the status of a User, being a Customer or not being a Customer, and the value of C_ID.

Legend: The boxes with rounded corners represent events that cause changes in state. Time generally flows down the page, in the sense that transitions that occur later in a User Session are shown further down the page. The term intr. is an abbreviation for web interaction.

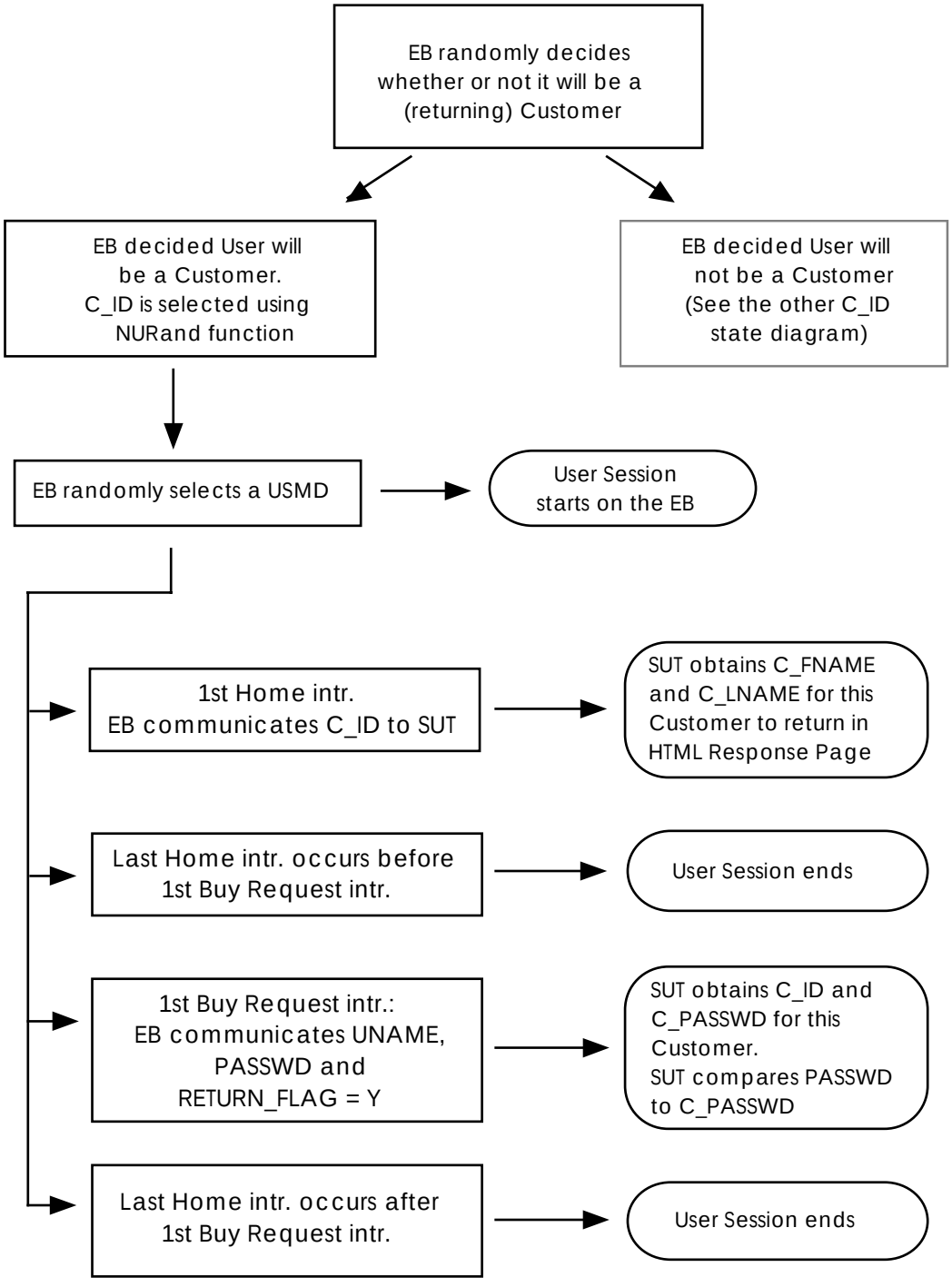
A.2 The State Transition Diagram 1

The following state transition diagram illustrates the situation when the User is **not** a Customer.



A.3 The State Transition Diagram 2

The following state transition diagram illustrates the situation when the User **is** a Customer.



Appendix B -- State Diagram of SHOPPING_ID

B.1 Introductory Explanations

The Specification says the following with respect to Shopping Session, SHOPPING_ID, and CART:

A SHOPPING_ID can be assigned to a User Session before a Shopping Session has been established. It might be helpful to think of a SHOPPING_ID as a web session ID used by the SUT to track the User Session. As such, it does not necessarily entail the existence of a Shopping Session. (It was deemed useful for the SUT to have the flexibility to track User Sessions using a SHOPPING_ID prior to creation of a CART.)

On the other hand, the existence of a Shopping Session requires both the assignment of a SHOPPING_ID and the creation of an initialized CART. A Shopping Session can be established earlier that the first Shopping Cart web interaction, but once the first Shopping Cart web interaction is executed, the Shopping Session *must* be established. (Clause 2.1.7.)

The state transition diagram that follows is an illustration of the above.

B.2 The State Values

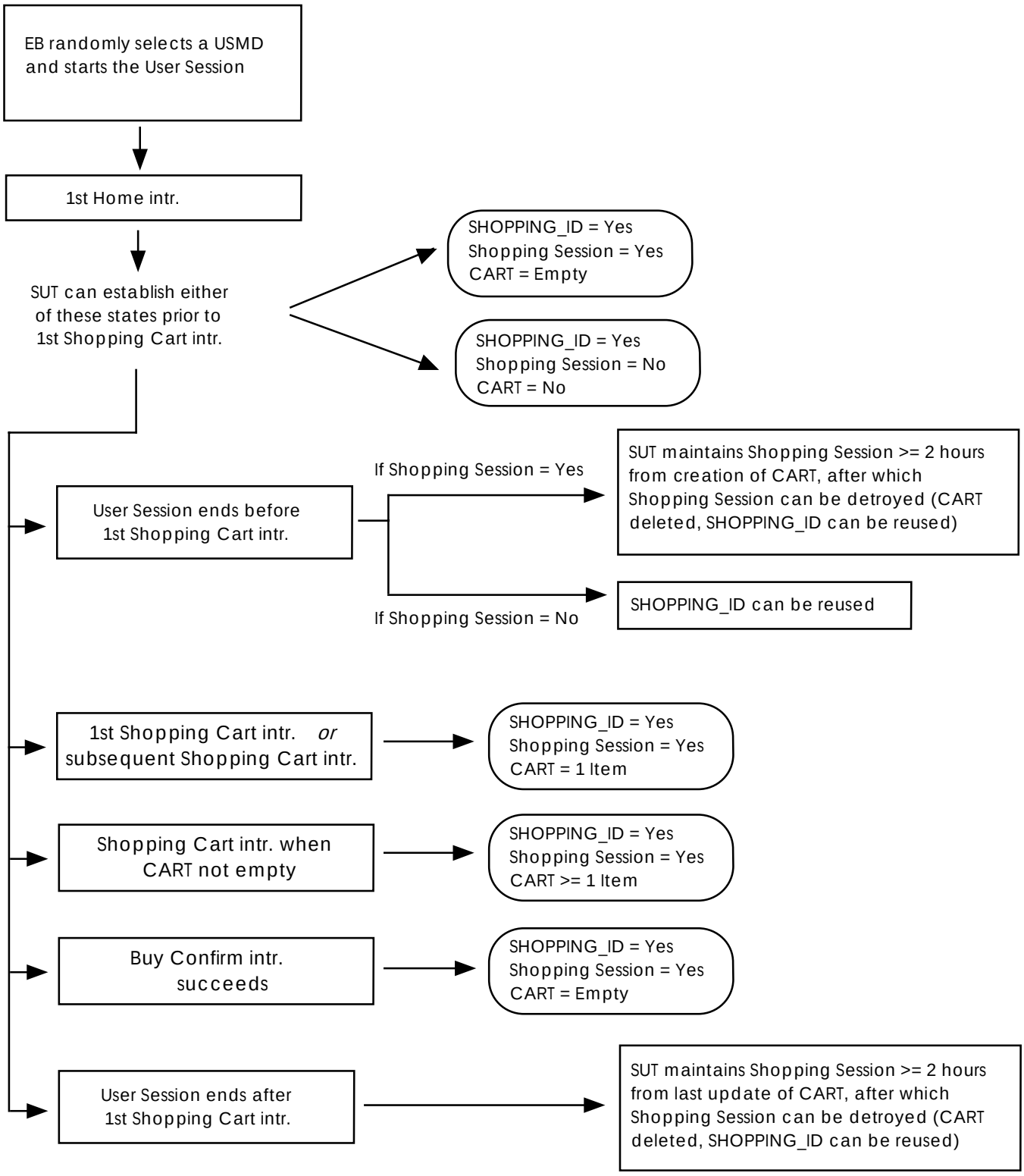
The diagram shows transitions in the states of three items associated with a User Session. These three items, along with how their states are represented in the diagram, are :

Item	Possible states of item	Meaning
SHOPPING_ID	SHOPPING_ID No	User Session does not have a SHOPPING_ID assigned.
	SHOPPING_ID Yes	User Session does have a SHOPPING_ID assigned.
Shopping Session	Shopping Session No	A Shopping Session has not been established.
	Shopping Session Yes	A Shopping Session has been established.
CART	CART No	A CART does not exist.
	CART Empty	An initialized CART exists with zero items in it.
	CART N items	A CART exists with N items in it.

The diagram shows the possible transitions in the state of these three items during a User Session. The rectangular (square cornered) boxes show possible states of these three items.

Legend: The boxes with rounded corners represent events that cause changes in state. Time generally flows down the page, in the sense that transitions that occur later in a User Session are shown further down the page. The term intr. is an abbreviation for web interaction.

B.3 The state Transition Diagram



B.4 Communication between RBE and SUT of existence of Shopping Session

The following information is not captured in the foregoing state transition diagram:

Once a Shopping Session has been established, Clause 2.1.7 requires that in all subsequent web interactions the RBE communicates to the SUT that a Shopping Session is active. Because the SUT has the flexibility to establish a Shopping Session prior to the first Shopping Cart web interaction of the User Session, the RBE, cannot depend on the execution of the first Shopping Cart web interaction as a trigger for it to start communicating to the SUT that a Shopping Session exists.

Therefore, the SUT must communicate to the RBE when a Shopping Session has been established. Once the RBE receives this information, it must communicate this same fact back to the SUT in all subsequent web interactions. In other words, it is the SUT, not the RBE, that determines when a Shopping Session has been started on behalf of the User. And it is the SUT that instructs the RBE on how to communicate the Shopping Session information back to the SUT during subsequent requests for web interactions. The exchange of a Cookie is an example of a mechanism that could be used for this purpose.

Appendix C -- Introduction to SSL

C.1 Overview

SSL is the acronym for Secure Socket Layer, a protocol developed by Netscape for managing secure (encrypted) communications in a network. Netscape has offered SSL as a proposed standard protocol to the World Wide Web Consortium (W3C) and the Internet Engineering Task Force (IETF) as a standard security approach for Web browsers and servers.

Netscape's SSL uses the public-and-private key encryption system from RSA, which also includes the use of a digital certificate for the server. RSA implemented public/private key encryption, called asymmetric encryption, which uses two unlike keys for secure communication. RSA has a patent on the RC2, RC4, RC5 ciphers which are supported by many browsers. A license fee must be paid to RSA for their use. This benchmark will implement the DES cipher for SSL, which does not require license fees.

The SSL protocol is initiated by a handshake wherein client and server exchange information and agree on ciphers. It is the handshake phase of the SSL communication, that makes use of the server's digital certificate. Optionally, the clients can also have digital certificates. However this benchmark does not require simulated clients (the EB's) to use digital certificates for SSL.

A digital certificate is an electronic "credit card" that establishes the holders credentials for doing business or other transactions on the Web. It is issued by a certification authority (CA). It contains his name, a serial number, expiration dates, a copy of the certificate holder's public key (used for encrypting and decrypting messages and digital signatures), and the digital signature of the certificate-issuing authority so that a recipient can verify that the certificate is real. Some digital certificates conform to a standard, X.509. Digital certificates can be kept in registries so that authenticated users can look up other users' public keys.

Digital certificates contain the holder's PUBLIC key. Typically, the certificate holder (in this case the server) sends his certificate to his partner (in this case the client). The client uses the server public key, found in the certificate, to encrypt information. Only the holder of the matching PRIVATE key can decrypt this data. Links to sample digital certificates can be found at the following web site: http://sitesearch.netscape.com/products/security/technology/certificates.html. Another good source of information is on the web at:

http://www.software.ibm.com/commerce/registry/primer.html

Data encrypted by the client during the SSL handshake actually forms the basis of the key that will be used with the selected cipher (e.g. DES) to encrypt the data content sent between client and server. Following the handshake and establishment of an SSL session, all secure web interactions flowing on this SSL session must be encrypted using the selected cipher and the session key established for this session.

C.2 SSL V3 Specifications

SSL V3 specifications are available at various web sites including the following (as of March 1999):

http://home.netscape.com/eng/ssl3/index.html

http://www.lanl.gov/projects/ia/stds/ia830810.html

Please note, URL references found in this document were accurate at the time it was written. But URL links are notoriously ephemeral. Should you discover the references to be invalid, please notify the TPC-W subcommittee and this document will be updated.

C.3 SSL V3 Implementation Toolkits

There are many versions of SSL toolkits and encryption libraries available to developers wishing to enable Web Servers for SSL or build browser emulation tools. Netscape's SSLRef program library can be downloaded for noncommercial use or licensed for commercial use.

The most popular SSL library is SSLeay, which can be downloaded from http://www.ssleay.org/. SSLeay is a free implementation of Secure Socket Layer.

SSLeay is pronounced S-S-L-e-a-y (i.e. each letter is pronounced individually).

SSLeay implements both SSLv2 (version 2) and SSLv3 (version 3) and TLSv1 as of the release of SSLeay-0.9.0. Documentation for building and using the library can be found at http://www.psy.uq.edu.au/~ftp/Crypto/ssleay/.

CryptSoft Pty Ltd acts as a focal point for consulting activities based around internet cryptography with specific focus on SSLeay and SSLapps.

C.4 Obtaining a Server Digital Certificate for Testing

Digital certificates are issued by a Certificate Authority (CA). They usually cost several hundred dollars and each Web Server host machine usually has its own certificate. The host name and domain name are encoded in the certificate. Using a certificate for a different server generally does not work.

There are several methods for obtaining a digital certificate for testing purposes without incurring the license fee. Certificate Authorities will generally provide a free test certificate, valid for some short time period of time. For example, you can obtain a free certificate, valid for two weeks, from Verisign Corp. Also Thawte provides free digital certificates for testing. See the following web site: https://www.thawte.com/cgi/server/test.exe. Some other Certificate Authorities are GTE-CyberTrust, Keywitness, and Tradewave.

An easier method for obtaining a test certificate is to build it yourself. This can be accomplished with the SSLeay toolkit. See http://www.intertrader.com/library/SSLeay/no_rsa.cfm#ownca. Many Web Servers also have the ability to act as a Certificate Authority and self-certify their own certificates. This is not recommended for production systems, but it is useful for testing. Read the documentation for your particular Web Server.

Intel offers Digital Certificate Manager software you can download. The Digital Certificate Manager is a sample application that allows end-users to view existing digital certificates and create new digital certificates. The certificate manager displays standard name and address fields present in

most types of certificates, as well as graphical data such as a photo of the certificate owner or a digitized cursive signature of that owner. See:

http://www.intel.com/ial/security/cdsa/dcm.htm

Appendix D -- Database Population Rationale

D.1 Derivation of 2880 Customers per Browser

Think time per interaction	=	7 seconds (with negative exponential distribution)
Response time per interaction	=	3 seconds avg. 90% < 20 seconds
Time per session	=	15 minutes
Measurement Interval	=	30 minutes
Sessions per measurement Interval	=	2
Types of Measurement intervals	=	WIPS, WIPSb, WIPSo

0.1428 interactions per second per concurrent browser

Customers	Browser Hours	30 Day Total Customers
4	1	2,880
40	10	28,800
400	100	288,000

Using 30 days of user information:

30 days x 24 hours x 4 sessions per browser hour = 2880 customer interactions per hour

The cardinality of customer table is: 2880 * (number. of concurrent browsers)

Therefore, 1,000 concurrent browsers yields:

2,880,000 customer table rows

Maximum of 142.8 WIPS ((1/7 interaction / think time) * 1,000)

Average of 100.0 WIPS ((1 / (7+3) interaction / (think time + response time)) * 1,000)

D.2 Using 8 hrs of Peak Throughput to Compute Storage

Even though the environment being modeled is a 24 hour operation, the traffic to the site is not uniformly distributed throughout the day. It typically goes through a couple of hours of peak traffic followed by a lull, to be picked up again, and so forth. Feedback from existing sites indicate peak throughput to be 2.6 to 3.3 times the average throughput. This is equivalent to 9.2 to 7.2 hours of peak throughput. Therefore, the 8hrs of peak throughput is an attempt at modeling the volume fluctuation typically seen in a 24 hour environment.

Appendix E -- Sample Pseudo-Code

Home web interaction	Select DISTINCT C_FNAME,C_LNAME from CUSTOMER where C_UNAME=@C_UNAME
New Products web interaction	select top 50 I_ID,I_TITLE,A_FNAME,A_LNAME from ITEM , AUTHOR where A_ID = I_A_ID AND I_SUBJECT LIKE @CategoryID order by I_PUB_DATE desc,I_TITLE
Best Sellers web interaction	select @last_o = max(OL_O_ID)-3333 from ORDER_LINE select top 50 I_ID,I_TITLE,A_FNAME,A_LNAME,SUM(OL_QTY) as TOTAL from ITEM, AUTHOR ,ORDER_LINE where OL_O_ID > @last_o AND I_ID = OL_I_ID AND I_A_ID = A_ID AND I_SUBJECT = @CategoryID group by I_ID,I_TITLE,A_FNAME,A_LNAME order by TOTAL desc
Product Detail web interaction	SELECT DISTINCT * FROM ITEM,AUTHOR WHERE AUTHOR.A_ID = ITEM.I_A_ID AND ITEM.I_ID = @BookID
Search Result web interaction (by Author)	SELECT top 50 I_TITLE,I_ID,A_FNAME, A_LNAME FROM ITEM,AUTHOR WHERE (I_A_ID = A_ID AND A_LNAME LIKE '%' + @Author + '%') order by I_TITLE
Search Result web interaction (by Title)	SELECT TOP 50 I_TITLE, I_ID, A_FNAME, A_LNAME FROM ITEM, AUTHOR WHERE I_A_ID = A_ID AND I_TITLE LIKE '%' + @Title + '%' order by I_TITLE
Search Result web interaction (by Subject)	SELECT TOP 50 I_TITLE,I_ID, A_FNAME, A_LNAME FROM ITEM, AUTHOR WHERE I_A_ID = A_ID AND I_SUBJECT LIKE '%' + @CategoryID + '%' order by I_TITLE
Order Display web interaction	select C_ID from CUSTOMER where C_UNAME=@C_UNAME and C_PASSWD- @C_PASSWD DECLARE @O_ID numeric(10) select @O_ID = max(O_ID)from ORDERS where O_C_ID=@C_ID SELECT C_FNAME,C_LNAME,C_EMAIL,C_PHONE, O_ID,O_DATE,O_SUBTOTAL,O_TAX,O_TOTAL,O_SHIP_TYPE,O_SHIP_DAT E, O_BILL_ADDR,O_SHIP_ADDR,O_CC_TYPE,O_STATUS, ADDR_STREET1,ADDR_STREET2,ADDR_CITY,ADDR_STATE,ADDR_ZIP,CO _NAME FROM CUSTOMER,ADDRESS,COUNTRY,ORDERS

	where O_ID=@O_ID and C_ID=@C_ID and O_BILL_ADDR=ADDR_ID AND ADDR_CO_ID=CO_ID Select ADDR_STREET1,ADDR_STREET2,ADDR_CITY, ADDR_STATE,ADDR_ZIP,CO_NAME from ADDRESS,COUNTRY where ADDR_ID=@A_ID and ADDR_CO_ID=CO_ID select OL_I_ID,I_TITLE,I_PUBLISHER, I_COST,OL_QTY,OL_DISCOUNT,OL_COMMENTSı from ORDER_LINE,ITEM where OL_I_ID=I_ID and OL_O_ID=@O_ID
Shopping Cart web interaction (Add to Cart)	Insert into SHOPPING_CART values(@Session,0,@BookID,1, @Title,@SRP,@COST,@Backing,GetDate())
Shopping Cart web interaction (Refresh Display)	Update SHOPPING_CART set SC_QTY=@QTY where SC_ID=@UserID and SC_I_ID=@BookID Delete from SHOPPING_CART where SC_ID=@UserID and SC_I_ID=@BookID
Buy Request web interaction (New Customer)	DECLARE @num numeric(10) SELECT @num=count(*) FROM CUSTOMER WHERE C_UNAME = @CustID select @num DECLARE @CO_ID numeric(4) DECLARE @ADDR_ID numeric(10) Select @CO_ID = CO_ID from COUNTRY where CO_NAME=@CO_NAME SELECT ADDR_ID FROM ADDRESS WHERE ADDR_STREET1=@ADDR_STREET1 and ADDR_STREET2=@ADDR_STREET2 and ADDR_CITY=@ADDR_CITY and ADDR_STATE=@ADDR_STATE and ADDR_ZIP=@ADDR_ZIP and ADDR_CO_ID = @CO_ID DECLARE @CO_ID numeric(4) DECLARE @ADDR_ID numeric(10) DECLARE @CUST_ID numeric(9) Select @CO_ID = CO_ID from COUNTRY where CO_NAME=@CO_NAME Insert into ADDRESS values(@ADDR_STREET1,@ADDR_STREET2,@ADDR_CITY,@ADDR_STATE, @ADDR_ZIP,@CO_ID) select @ADDR_ID = @@identity

	Insert into CUSTOMER values(@C_UNAME,@C_PASSWD,@C_FNAME,@C_LNAME,@ADDR_ID,@C_ PHONE,@C_EMAIL,getdate(),getdate(),NULL,NULL,@C_DISCOUNT,0,0,@C_BI RHTDAY,@C_DATA1,@C_DATA2) select @CUST_ID = @@identity select @CUST_ID
Buy Request web interaction (Existing Customer)	select C_ID from CUSTOMER where C_UNAME=@C_UNAME and C_PASSWD=@C_PASSWD SELECT C_UNAME,C_PASSWD,C_FNAME,C_LNAME,C_PHONE, C_EMAIL,C_BIRTHDATE,C_DATA1,C_DATA2,ADDR_STREET1, ADDR_STREET2,ADDR_CITY,ADDR_STATE,ADDR_ZIP,CO_NAME FROM CUSTOMER,ADDRESS,COUNTRY where C_ADDR_ID=ADDR_ID and ADDR_CO_ID=CO_ID and C_ID = @C_ID
Buy Confirm web interaction	DECLARE @CO_ID numeric(4) DECLARE @ADDR_ID numeric(10) Select @CO_ID = CO_ID from COUNTRY where CO_NAME=@CO_NAME SELECT ADDR_ID FROM ADDRESS WHERE ADDR_STREET1=@ADDR_STREET1 and ADDR_STREET2=@ADDR_STREET2 and ADDR_CITY=@ADDR_CITY and ADDR_STATE=@ADDR_STATE and ADDR_ZIP=@ADDR_ZIP and ADDR_CO_ID=@CO_ID Select @CO_ID = CO_ID from COUNTRY where CO_NAME=@CO_NAME Insert into ADDRESS values(@ADDR_STREET1, @ADDR_STREET2,@ADDR_CITY,@ADDR_STATE,@ADDR_ZIP, @CO_ID) select @ADDR_ID = @@identity select C_ID,C_DISCOUNT,C_ADDR_ID from CUSTOMER where C_UNAME=@UserID DECLARE @O_ID numeric(9) Insert into ORDERS values (@O_C_ID,getdate(),@O_SUBTOTAL, @O_TAX,@O_TOTAL,@O_SHIP_TYPE,NULL,@O_BILL_ADDR, @O_SHIP_ADDR,@O_CC_TYPE,@O_CC_NUM,@O_CC_NAME, @O_CC_EXP,'Pending') select @O_ID = @@identity Insert ORDER_LINE (OL_O_ID,OL_I_ID,OL_QTY, OL_DISCOUNT,OL_COMMENTS)

	Select @O_ID,SC_I_ID,SC_QTY,1,'comment' from SHOPPING_CART Delete from SHOPPING_CART where SC_ID=@Session Insert into CC_XACTS values(@O_ID,@O_CC_TYPE,@O_CC_EXP,@O_CC_AUTH,@O_TOTAL,getdate(),@CO_ID) Select @O_ID
Admin Request web interaction	SELECT DISTINCT * FROM ITEM,AUTHOR WHERE AUTHOR.A_ID = ITEM.I_A_ID AND ITEM.I_ID = @BookID
Admin Confirm web interaction	Update ITEM set I_COST=@Cost,I_PUB_DATE = @Dait, I_IMAGE = @Gif_Loc where I_ID=@Item_ID

Appendix F -- Sample HTML Code

F.1 Admin Confirm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN">
<HTML>
<HEAD>
<TITLE>TPC-W Admin Response Page</TITLE></HEAD>
<BODY BGCOLOR="#FFFFFF">
<H1 ALIGN="CENTER">TPC Web Commerce Benchmark (TPC-W)</H1>
<H2 ALIGN="CENTER"><IMG
SRC="tpclogo.gif" ALIGN="BOTTOM" BORDER="0" WIDTH="288" HEIGHT="67"></H2>
<H2 ALIGN="CENTER">Admin Confirm Page</H2>
<H2>Product Updated</H2>
<H2>Title:bwRKVvI2uoUsaraNxYXJzUQhKQDBASEALINOGeJzOAO36rsR7F2</H2>
<P>Author: AFG2tt3ug L1kJ3z3mpBmXJi</P>
<P><IMG SRC="50k.gif" ALIGN="RIGHT" BORDER="0" WIDTH="200" HEIGHT="200"><IMG
SRC="5kgif2.gif" ALT="Book 1" ALIGN="RIGHT" WIDTH="100" HEIGHT="150">Description:
nkEkKMUGtPyhnwz7Ph83k5YIf95ElporR6Yz7KJ6w G9f3nBsmBaUeRcpe6f8xD0rYH4VykyA
Q3ji7ix6GjQoby2Z5z7vu9OzMrMENVjEDNdYTGkMe Se3hDgeEZpPqykynd7yepX2a7 eSexHb
fGYG9YOEQOppbXt6sYdzkWr</P>
<BLOCKQUOTE>
<P><B>Suggested Retail: $297.60</B><BR><B>Our Price:</B><FONT
COLOR="#DD0000"><B> $222.33</B></FONT><BR><B>You Save:</B><FONT
COLOR="#DD0000"><B> $75.27</B></FONT></P></BLOCKQUOTE>
<P><FONT SIZE="2">PAPERBACK, 45651 pages<BR>Published by
EuDtr0vomtI8jWDIe0c84lovtXiNfgnIuCEfPMWIAkAdzdd1j5pMYSTERY<BR>Publication date:
7-15-1992<BR>Dimensions (in inches): 55.71 x 13.47 x 48.51<BR>ISBN:
2xXwLBDueBZ5b</FONT><BR CLEAR="ALL"></P>
<P ALIGN="CENTER">
<A HREF="tpcw.dll?search&Session=2880021&C_ID=2880021"><IMG
SRC="search.gif" ALT="Search" BORDER="0" WIDTH="120" HEIGHT="30"></A>
<A HREF="tpcw.dll?home&Session=2880021&C_ID=2880021"><IMG
SRC="home.gif" ALT="Home" BORDER="0" WIDTH="120" HEIGHT="30"></A>
</P></BODY></HTML>
```

F.2 Admin Request

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN">
<HTML>
<HEAD>
<TITLE>TPC-W Product Update Page</TITLE></HEAD>
<BODY BGCOLOR="#ffffff">
<H1 ALIGN="center">TPC Web Commerce Benchmark (TPC-W)</H1>
<H2 ALIGN="center"><IMG
SRC="tpclogo.gif" ALIGN="BOTTOM" BORDER="0" WIDTH="288" HEIGHT="67"></H2>
<H2 ALIGN="center">Admin Request Page</H2>
<H2 ALIGN="center">Title:bwRKVvI2uoUsaraNxYXJzUQhKQDBASEALINOGeJzOAO36rsR7F2</H2>
<P ALIGN="LEFT">Author: AFG2tt3ug L1kJ3z3mpBmXJi<BR></P><IMG
SRC="25k.gif" ALIGN="RIGHT" BORDER="0" WIDTH="200" HEIGHT="200"><IMG
SRC="5kgif1.gif" ALIGN="RIGHT">
<P><BR><BR></P>
<FORM ACTION="tpcw.dll?ChangeInfo&amp;Session=0&amp;C_ID=0" METHOD="post"><INPUT
NAME="BookID" TYPE="hidden" VALUE="47177">
<TABLE BORDER="0">
<TR>
<TD><B>Suggested Retail:</B></TD>
<TD><B>$ 297.60</B></TD></TR>
<TR>
<TD><B>Our Current Price:</B></TD>
<TD><FONT COLOR="#dd0000"><B>$ 242.80</B></FONT></TD></TR>
<TR>
<TD><B>Enter New Price</B></TD>
<TD ALIGN="right">$                          <INPUT NAME="price"></TD></TR>
<TR>
<TD><B>Enter New Picture</B></TD>
```

```
<TD ALIGN="right"><INPUT NAME="pic"></TD></TR>
<TR>
<TD><B>Enter New Thumbnail</B></TD>
<TD ALIGN="RIGHT"><INPUT TYPE="TEXT" NAME="thumb"></TD></TR></TABLE>
<P><BR CLEAR="ALL"></P>
<P ALIGN="center"><INPUT
TYPE="IMAGE" NAME="Submit Changes" SRC="changes.gif" BORDER="0" HEIGHT="30"
WIDTH="130"><A
HREF="tpcw.dll?Search&Session=0&C_ID=0"><IMG
SRC="search.gif" ALT="Search" BORDER="0" HSPACE="0" WIDTH="120" HEIGHT="30"></A><A
HREF="tpcw.dll?home&Session=0&C_ID=0"><IMG
SRC="home.gif" ALT="Home" BORDER="0" HSPACE="0" WIDTH="120"
HEIGHT="30"></A></P></FORM></BODY></HTML>
```

F.3 Best Sellers

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN">
<HTML>
<HEAD>
<TITLE>Order Confirmation</TITLE></HEAD>
<BODY BGCOLOR="#FFFFFF">
<H1 ALIGN="CENTER">TPC Web Commerce Benchmark (TPC-W)</H1>
<H2 ALIGN="CENTER">Buy Confirm Page</H2>
<BLOCKQUOTE>
<BLOCKQUOTE>
<BLOCKQUOTE>
<BLOCKQUOTE>
<H2 ALIGN="LEFT">Order Information:</H2>
<TABLE BORDER="1" CELLSPACING="0" CELLPADDING="0">
<TR>
<TD><B>Qty</B></TD>
<TD><B>Product</B></TD></TR>
<TR>
<TD VALIGN="TOP">1</TD>
<TD VALIGN="TOP">Title:<I>1OGRENGSERIgVR</I> - Backing:HARDBACK <BR>SRP.
$986.25,               <FONT COLOR="#aa0000"><B>Your Price:
$269.83</B></FONT></TD></TR>
<TR>
<TD VALIGN="TOP">1</TD>
<TD VALIGN="TOP">Title:<I>KrsfIHwvioZuZOGINOGALSEM4WiQDCl</I> -
Backing:HARDBACK <BR>SRP. $701.72,
<FONT COLOR="#aa0000"><B>Your Price:              $500.17</B></FONT></TD></TR>
<TR>
<TD VALIGN="TOP">1</TD>
<TD VALIGN="TOP">Title:<I>axdxCeeRrJViu6Y0Xrr0jTq8BOGRIBAOGBAe1rAkUrk2RfeftySIE</I>
             - Backing:AUDIO <BR>SRP. $270.11,
<FONT COLOR="#aa0000"><B>Your Price:
$257.18</B></FONT></TD></TR></TABLE>
<H2 ALIGN="LEFT">Your Order has been processed.</H2>
<TABLE BORDER="1" CELLPADDING="5" CELLSPACING="0">
<TR>
<TD>
<H4>Subtotal with discount:</H4></TD>
<TD>
<H4>$719.03</H4></TD></TR>
<TR>
<TD>
<H4>Tax (8.25%):</H4></TD>
<TD>
<H4>$59.32</H4></TD></TR>
<TR>
<TD>
<H4>Shipping &amp;amp; Handling:</H4></TD>
<TD>
<H4>$6.00</H4></TD></TR>
<TR>
<TD>
<H4>Total:</H4></TD>
<TD>
<H4>$784.35</H4></TD></TR></TABLE>
```

```
<P><BR></P>
<H2>Order Number: 2592021</H2>
<H1>Thank you for shopping at TPC-W</H1>
<P></P>
<CENTER><A
HREF="http://fishbowl4/scripts/tpcw.dll?search&Session=5678&C_ID=2880005"><IMG
SRC="search.gif" ALT="Search Item" BORDER="0" VSPACE="0" HSPACE="0" WIDTH="120"
HEIGHT="30"></A><A
HREF="http://fishbowl4/scripts/tpcw.dll?home&Session=5678&C_ID=2880005"><IMG
SRC="home.gif" ALT="Home Page" BORDER="0" VSPACE="0" HSPACE="0" WIDTH="120"
HEIGHT="30"></A></CENTER></BLOCKQUOTE></BLOCKQUOTE></BLOCKQUOTE>
</BLOCKQUOTE></BODY></HTML>
```

F.4 Buy Confirm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN">
<HTML>
<HEAD>
<TITLE>Order Confirmation</TITLE></HEAD>
<BODY BGCOLOR="#FFFFFF">
<H1 ALIGN="CENTER">TPC Web Commerce Benchmark (TPC-W)</H1>
<H2 ALIGN="CENTER">Buy Confirm Page</H2>
<BLOCKQUOTE>
<BLOCKQUOTE>
<BLOCKQUOTE>
<BLOCKQUOTE>
<H2 ALIGN="LEFT">Order Information:</H2>
<TABLE BORDER="1" CELLSPACING="0" CELLPADDING="0">
<TR>
<TD><B>Qty</B></TD>
<TD><B>Product</B></TD></TR>
<TR>
<TD VALIGN="TOP">1</TD>
<TD VALIGN="TOP">Title:<I>1OGRENGSERIgVR</I> - Backing:HARDBACK -
Publisher:jksdguyFlhks<BR>SRP. $986.25,              <FONT COLOR="#aa0000"><B>Your
Price: $269.83</B></FONT></TD></TR>
<TR>
<TD VALIGN="TOP">1</TD>
<TD VALIGN="TOP">Title:<I>KrsfIHwvioZuZOGINOGALSEM4WiQDCl</I> -
Backing:HARDBACK - Publisher:YjoqwHFlfhj<BR>SRP. $701.72,
<FONT COLOR="#aa0000"><B>Your Price:              $500.17</B></FONT></TD></TR>
<TR>
<TD VALIGN="TOP">1</TD>
<TD VALIGN="TOP">Title:<I>axdxCeeRrJViu6Y0Xrr0jTq8BOGRIBAOGBAe1rAkUrk2RfeftySIE</I>
              - Backing:AUDIO - Publisher:HKigjfTYdjTJK<BR>SRP. $270.11,
<FONT COLOR="#aa0000"><B>Your Price:
$257.18</B></FONT></TD></TR></TABLE>
<H2 ALIGN="LEFT">Your Order has been processed.</H2>
<TABLE BORDER="1" CELLPADDING="5" CELLSPACING="0">
<TR>
<TD>
<H4>Subtotal with discount:</H4></TD>
<TD>
<H4>$719.03</H4></TD></TR>
<TR>
<TD>
<H4>Tax (8.25%):</H4></TD>
<TD>
<H4>$59.32</H4></TD></TR>
<TR>
<TD>
<H4>Shipping &amp;amp; Handling:</H4></TD>
<TD>
<H4>$6.00</H4></TD></TR>
<TR>
<TD>
<H4>Total:</H4></TD>
<TD>
<H4>$784.35</H4></TD></TR></TABLE>
<P><BR></P>
```

```
<H2>Order Number: 2592021</H2>
<H1>Thank you for shopping at TPC-W</H1>
<P></P>
<CENTER><A
HREF="http://fishbowl4/scripts/tpcw.dll?search&Session=5678&C_ID=2880005"><IMG
SRC="search.gif" ALT="Search Item" BORDER="0" VSPACE="0" HSPACE="0" WIDTH="120"
HEIGHT="30"></A><A
HREF="http://fishbowl4/scripts/tpcw.dll?home&Session=5678&C_ID=2880005"><IMG
SRC="home.gif" ALT="Home Page" BORDER="0" VSPACE="0" HSPACE="0" WIDTH="120"
HEIGHT="30"></A></CENTER></BLOCKQUOTE></BLOCKQUOTE></BLOCKQUOTE></BLOCKQUOTE>
</BODY></HTML>
```

F.5 Buy Request

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN">
<HTML>
<HEAD>
<TITLE>TPC-W Buy Request</TITLE></HEAD>
<BODY BGCOLOR="ffffff">
<H1 ALIGN="CENTER">TPC Web Commerce Benchmark (TPC-W)</H1>
<H2 ALIGN="CENTER">Buy Request Page</H2>
<HR>
<FORM ACTION="tpcw.dll?BuyConfirm&Session=5678&C_ID=2880005" METHOD="POST">
<TABLE BORDER="0" WIDTH="90%">
<TR ALIGN="LEFT" VALIGN="TOP">
<TD VALIGN="TOP" WIDTH="45%">
<H2>Billing Information:</H2>
<TABLE WIDTH="100%" BORDER="0">
<TR>
<TD>Firstname:</TD>
<TD> John</TD></TR>
<TR>
<TD>Lastname: </TD>
<TD>Doe</TD></TR>
<TR>
<TD>Addr_street_1:</TD>
<TD> 1 Some Place</TD></TR>
<TR>
<TD>Addr_street_2:</TD>
<TD> Apt. 42</TD></TR>
<TR>
<TD>City:</TD>
<TD> Dullsville</TD></TR>
<TR>
<TD>State:</TD>
<TD> CA</TD></TR>
<TR>
<TD>Zip:</TD>
<TD>91234</TD></TR>
<TR>
<TD>Country:</TD>
<TD> Andorra</TD></TR>
<TR>
<TD>Email:</TD>
<TD> me@mycompany.com</TD></TR>
<TR>
<TD>Phone:</TD>
<TD> 123-456-7890</TD></TR>
<TR>
<TD>USERNAME:</TD>
<TD>ALININBABABASE</TD></TR></TABLE></TD>
<TD VALIGN="TOP" WIDTH="45%">
<H2>Shipping Information:</H2>
<TABLE BORDER="0" CELLSPACING="0" CELLPADDING="0" WIDTH="100%">
<TR>
<TD WIDTH="50%">Addr_street_1:</TD>
<TD WIDTH="50%"><INPUT NAME="shipadd1" SIZE="40" VALUE="1 Some Place"></TD></TR>
<TR>
<TD>Adde_street_ 2:</TD>
<TD><INPUT NAME="shipadd2" SIZE="40" VALUE="Apt. 42"></TD></TR>
```

```
<TR>
<TD>City:</TD>
<TD><INPUT NAME="shipcity" SIZE="30" VALUE="Dullsville"></TD></TR>
<TR>
<TD>State:</TD>
<TD><INPUT NAME="shipstate" SIZE="20" VALUE="CA"></TD></TR>
<TR>
<TD>Zip:</TD>
<TD><INPUT NAME="shipzip" SIZE="10" VALUE="91234"></TD></TR>
<TR>
<TD>Country:</TD>
<TD><INPUT NAME="shipcountry" VALUE="Andorra"
SIZE="40"></TD></TR></TABLE></TD></TR></TABLE>
<HR>
<H2>Order Information:</H2>
<TABLE BORDER="1" CELLSPACING="0" CELLPADDING="0">
<TR>
<TD><B>Qty</B></TD>
<TD><B>Product</B></TD></TR>
<TR>
<TD VALIGN="TOP">1</TD>
<TD VALIGN="TOP">Title:<I>1OGRENGSERIgVR</I> - Backing:HARDBACK <BR>SRP.
$986.25,              <FONT COLOR="#aa0000"><B>Your Price:
$269.83</B></FONT></TD></TR></TABLE>
<P><BR></P>
<TABLE BORDER="0">
<TR>
<TD><B>Subtotal with discount: </B></TD>
<TD ALIGN="RIGHT"><B>$188.88</B></TD></TR>
<TR>
<TD><B>Tax</B></TD>
<TD ALIGN="RIGHT"><B>$15.58</B></TD></TR>
<TR>
<TD><B>Shipping &amp;amp; Handling</B></TD>
<TD ALIGN="RIGHT"><B>$4.00</B></TD></TR>
<TR>
<TD><B>Total</B></TD>
<TD ALIGN="RIGHT"><B>$478.29</B></TD></TR></TABLE>
<HR>
<P><BR></P>
<TABLE BORDER="1" CELLPADDING="5" CELLSPACING="0">
<TR>
<TD>Credit Card Type</TD>
<TD><INPUT TYPE="RADIO" NAME="cctype" VALUE="Visa" CHECKED="CHECKED">Visa<INPUT
TYPE="RADIO" NAME="cctype" VALUE="Master">MasterCard<INPUT
TYPE="RADIO" NAME="cctype" VALUE="Dinner">Dinners<INPUT
TYPE="RADIO" NAME="cctype" VALUE="Amex">American Express</TD></TR>
<TR>
<TD>Name on Credit Card</TD>
<TD><INPUT NAME="ccname" SIZE="30" VALUE="John Doe"></TD></TR>
<TR>
<TD>Credit Card Number</TD>
<TD><INPUT NAME="ccnum" SIZE="16" VALUE="1234567890123456"></TD></TR>
<TR>
<TD>Credit Card Expiration Date</TD>
<TD><INPUT NAME="ccdate" SIZE="15" VALUE="02/02/1999"></TD></TR>
<TR>
<TD>Shipping Method</TD>
<TD><INPUT TYPE="RADIO" NAME="shipmethod" VALUE="Regular"
CHECKED="CHECKED">Regular<INPUT
TYPE="RADIO" NAME="shipmethod" VALUE="2-day">2-day<INPUT
TYPE="RADIO" NAME="shipmethod" VALUE="Overnight">Overnight</TD></TR></TABLE>
<P ALIGN="CENTER"><INPUT
TYPE="IMAGE" NAME="Confirm Buy" SRC="process.gif" HEIGHT="30" WIDTH="120"><A
HREF="http://fishbowl4/scripts/tpcw.dll?ShoppingBasket&ADD_FLAG=N&Session=5678&C_ID
=2880005"><IMG
SRC="cart.gif" ALT="Shopping Cart" BORDER="0" VSPACE="0" HSPACE="0" WIDTH="120"
HEIGHT="30"></A><A
HREF="http://fishbowl4/scripts/tpcw.dll?home&Session=5678&C_ID=2880005"><IMG
SRC="home.gif" ALT="Home" BORDER="0" VSPACE="0" HSPACE="0" WIDTH="120"
HEIGHT="30"></A></P></FORM></BODY></HTML>
```

F.6 Customer Registration

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN">
<HTML>
<HEAD>
<TITLE>Customer Registration</TITLE></HEAD>
<BODY BGCOLOR="#ffffff">
<H1 ALIGN="center">TPC Web Commerce Benchmark (TPC-W)</H1>
<H1 ALIGN="center"><IMG
SRC="tpclogo.gif" ALIGN="BOTTOM" BORDER="0" WIDTH="288" HEIGHT="67"></H1>
<H2 ALIGN="center">Customer Registration Page</H2>
<FORM ACTION="tpcw.dll?UserType&amp;Session=5678&amp;C_ID=0" METHOD="post">
<BLOCKQUOTE>
<BLOCKQUOTE>
<HR>
<TABLE BORDER="0">
<TR>
<TD><INPUT CHECKED="CHECKED" NAME="RETURNING_FLAG" TYPE="radio" VALUE="Y"> I
am an existing customer             </TD></TR>
<TR>
<TD><INPUT NAME="RETURNING_FLAG" TYPE="radio" VALUE="N"> I am a first time
customer</TD></TR></TABLE>
<HR>
<P><B>If you’re an existing customer, enter your User ID and
Password:</B><BR><BR></P>
<TABLE>
<TR ALIGN="left">
<TD>User ID:                      <INPUT NAME="username" SIZE="23"></TD></TR>
<TR ALIGN="left">
<TD>Password:
<INPUT SIZE="14" TYPE="password" VALUE="password"></TD></TR></TABLE>
<HR>
<P><B>If you’re a first time customer, enter the details below:</B><BR>
  </P>
<TABLE>
<TR>
<TD>Enter your birth date (mm/dd/yyyy):</TD>
<TD>                    <INPUT NAME="birthday" SIZE="10"></TD></TR>
<TR>
<TD>Enter your First Name:</TD>
<TD>                    <INPUT NAME="firstname" SIZE="15"></TD></TR>
<TR>
<TD>Enter your Last Name:</TD>
<TD>                    <INPUT NAME="lastname" SIZE="15"></TD></TR>
<TR>
<TD>Enter your Address 1:</TD>
<TD>                    <INPUT NAME="address1" SIZE="40"></TD></TR>
<TR>
<TD>Enter your Address 2:</TD>
<TD>                    <INPUT NAME="address2" SIZE="40"></TD></TR>
<TR>
<TD>Enter your City, State, Zip:</TD>
<TD>                    <INPUT NAME="city" SIZE="30">
<INPUT NAME="state">                      <INPUT NAME="zip" SIZE="10"></TD></TR>
<TR>
<TD>Enter your Country:</TD>
<TD>                    <INPUT NAME="country" SIZE="50"></TD></TR>
<TR>
<TD>Enter your Phone:</TD>
<TD>                    <INPUT NAME="phone" SIZE="16"></TD></TR>
<TR>
<TD>Enter your E-mail:</TD>
<TD>                    <INPUT NAME="email" SIZE="50"></TD></TR></TABLE>
<HR>
<TABLE>
<TR>
<TD COLSPAN="2">Special Instructions:
<TEXTAREA COLS="65" NAME="data" ROWS="4"></TEXTAREA></TD></TR></TABLE>
<HR></BLOCKQUOTE></BLOCKQUOTE>
<CENTER><INPUT
TYPE="IMAGE" NAME="Enter Order" SRC="submit.gif" BORDER="0" HEIGHT="30"
WIDTH="120"><A
```

```
HREF="http://fishbowl4/scripts/tpcw.dll?search&Session=5678&C_ID=0"><IMG
SRC="search.gif" ALT="Search Item" BORDER="0" HSPACE="0" WIDTH="120"
HEIGHT="30"></A><A
HREF="http://fishbowl4/scripts/tpcw.dll?home&Session=5678&C_ID=0"><IMG
SRC="home.gif" ALT="Home Page" BORDER="0" HSPACE="0" WIDTH="120" HEIGHT="30"></A>
<P></P></CENTER></FORM></BODY></HTML>
```

F.7 Home

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN">
<HTML>
<HEAD>
<TITLE>TPC-W Home Page</TITLE></HEAD>
<BODY BGCOLOR="#ffffff">
<H1 ALIGN="center">TPC Web Commerce Benchmark (TPC-W)</H1>
<P ALIGN="CENTER"><IMG
SRC="tpclogo.gif" ALIGN="BOTTOM" BORDER="0" WIDTH="288" HEIGHT="67"></P>
<H2 ALIGN="center">Home Page</H2>
<H3 ALIGN="center">Welcome back <FONT COLOR="#770000">John Doe</FONT></H3>
<TABLE ALIGN="CENTER" BORDER="0" WIDTH="660">
<TR ALIGN="CENTER" VALIGN="top">
<TD COLSPAN="5"><B><FONT COLOR="#ff0000" SIZE="+1">Click on one of our latest
books to find out more !</FONT></B></TD></TR>
<TR ALIGN="CENTER" VALIGN="top">
<TD><A HREF="tpcw.dll?BookInfo=59227&Session=6250&C_ID=0"><IMG
SRC="5kgif1.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=24568&Session=6250&C_ID=0"><IMG
SRC="5kgif2.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=94429&Session=6250&C_ID=0"><IMG
SRC="5kgif3.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=68639&Session=6250&C_ID=0"><IMG
SRC="5kgif4.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=49145&Session=6250&C_ID=0"><IMG
SRC="5kgif5.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD></TR></TABLE>
<TABLE
ALIGN="center" BGCOLOR="#c0c0c0" BORDER="0" CELLPADDING="6" CELLSPACING="0"
WIDTH="700">
<TR ALIGN="CENTER" BGCOLOR="#ffffff" VALIGN="top">
<TD COLSPAN="2" VALIGN="MIDDLE" WIDTH="300"><IMG
SRC="new.gif" ALT="New Product" WIDTH="300"></TD>
<TD BGCOLOR="#ffffff" WIDTH="100"></TD>
<TD COLSPAN="2" WIDTH="300"><IMG SRC="best.gif" ALT="Best Seller"
WIDTH="300"></TD></TR>
<TR>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=ARTS&Session=6250&C_ID=0">ARTS</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=NON-FICTION&Session=6250&C_ID=0">NON-
FICTION</A></P></TD>
<TD BGCOLOR="#ffffff" WIDTH="50"></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=ARTS&Session=6250&C_ID=0">ARTS</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=NON-FICTION&Session=6250&C_ID=0">NON-
FICTION</A></P></TD></TR>
<TR>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=BIOGRAPHIES&Session=6250&C_ID=0">BIOGRAPHIES</A></P></TD
>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=PARENTING&Session=6250&C_ID=0">PARENTING</A></P></TD>
<TD BGCOLOR="#ffffff" WIDTH="50"></TD>
<TD>
<P ALIGN="center"><A
```

```
HREF="tpcw.dll?best&subject=BIOGRAPHIES&Session=6250&C_ID=0">BIOGRAPHIES</A></P></T
D>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=PARENTING&Session=6250&C_ID=0">PARENTING</A></P></TD></
TR>
<TR>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=BUSINESS&Session=6250&C_ID=0">BUSINESS</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=POLITICS&Session=6250&C_ID=0">POLITICS</A></P></TD>
<TD BGCOLOR="#ffffff" WIDTH="50"></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=BUSINESS&Session=6250&C_ID=0">BUSINESS</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=POLITICS&Session=6250&C_ID=0">POLITICS</A></P></TD></TR
>
<TR>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=CHILDREN&Session=6250&C_ID=0">CHILDREN</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=REFERENCE&Session=6250&C_ID=0">REFERENCE</A></P></TD>
<TD BGCOLOR="#ffffff" WIDTH="50"></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=CHILDREN&Session=6250&C_ID=0">CHILDREN</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=REFERENCE&Session=6250&C_ID=0">REFERENCE</A></P></TD></
TR>
<TR>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=COMPUTERS&Session=6250&C_ID=0">COMPUTERS</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=RELIGION&Session=6250&C_ID=0">RELIGION</A></P></TD>
<TD BGCOLOR="#ffffff" WIDTH="50"></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=COMPUTERS&Session=6250&C_ID=0">COMPUTERS</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=RELIGION&Session=6250&C_ID=0">RELIGION</A></P></TD></TR
>
<TR>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=COOKING&Session=6250&C_ID=0">COOKING</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=ROMANCE&Session=6250&C_ID=0">ROMANCE</A></P></TD>
<TD BGCOLOR="#ffffff" WIDTH="50"></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=COOKING&Session=6250&C_ID=0">COOKING</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=ROMANCE&Session=6250&C_ID=0">ROMANCE</A></P></TD></TR>
<TR>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=HEALTH&Session=6250&C_ID=0">HEALTH</A></P></TD>
<TD>
<P ALIGN="center"><A HREF="tpcw.dll?new&subject=SELF-
HELP&Session=6250&C_ID=0">SELF-HELP</A></P></TD>
<TD BGCOLOR="#ffffff" WIDTH="50"></TD>
```

```
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=HEALTH&Session=6250&C_ID=0">HEALTH</A></P></TD>
<TD>
<P ALIGN="center"><A HREF="tpcw.dll?best&subject=SELF-
HELP&Session=6250&C_ID=0">SELF-HELP</A></P></TD></TR>
<TR>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=HISTORY&Session=6250&C_ID=0">HISTORY</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=SCIENCE-NATURE&Session=6250&C_ID=0">SCIENCE-
NATURE</A></P></TD>
<TD BGCOLOR="#ffffff" WIDTH="50"></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=HISTORY&Session=6250&C_ID=0">HISTORY</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=SCIENCE-NATURE&Session=6250&C_ID=0">SCIENCE-
NATURE</A></P></TD></TR>
<TR>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=HOME&Session=6250&C_ID=0">HOME</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=SCIENCE-FICTION&Session=6250&C_ID=0">SCIENCE-
FICTION</A></P></TD>
<TD BGCOLOR="#ffffff" WIDTH="50"></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=HOME&Session=6250&C_ID=0">HOME</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=SCIENCE-FICTION&Session=6250&C_ID=0">SCIENCE-
FICTION</A></P></TD></TR>
<TR>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=HUMOR&Session=6250&C_ID=0">HUMOR</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=SPORTS&Session=6250&C_ID=0">SPORTS</A></P></TD>
<TD BGCOLOR="#ffffff" WIDTH="50"></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=HUMOR&Session=6250&C_ID=0">HUMOR</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=SPORTS&Session=6250&C_ID=0">SPORTS</A></P></TD></TR>
<TR>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=LITERATURE&Session=6250&C_ID=0">LITERATURE</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?new&subject=MYSTERY&Session=6250&C_ID=0">MYSTERY</A></P></TD>
<TD BGCOLOR="#ffffff" WIDTH="50"></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=LITERATURE&Session=6250&C_ID=0">LITERATURE</A></P></TD>
<TD>
<P ALIGN="center"><A
HREF="tpcw.dll?best&subject=MYSTERY&Session=6250&C_ID=0">MYSTERY</A></P></TD></TR><
/TABLE>
<P ALIGN="CENTER"><A
HREF="tpcw.dll?ShoppingBasket&ADD_FLAG=N&Session=6250&C_ID=0"><IMG
SRC="cart.gif" ALT="Shopping Cart" BORDER="0" WIDTH="120" HEIGHT="30"></A><A
HREF="tpcw.dll?search&Session=6250&C_ID=0"><IMG
SRC="search.gif" ALT="Search" BORDER="0" WIDTH="120" HEIGHT="30"></A><A
HREF="tpcw.dll?OrderStatus&Session=6250&C_ID=0"><IMG
```

```
SRC="status.gif" ALT="Order Status" BORDER="0" WIDTH="120"
HEIGHT="30"></A></P></BODY></HTML>
```

F.8 New Products

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN">
<HTML>
<HEAD>
<TITLE>New ARTS</TITLE></HEAD>
<BODY BGCOLOR="#ffffff">
<H1 ALIGN="center">TPC Web Commerce Benchmark (TPC-W)</H1>
<P ALIGN="center"><IMG
SRC="tpclogo.gif" ALIGN="BOTTOM" BORDER="0" WIDTH="288" HEIGHT="67">
</P>
<P></P>
<H2 ALIGN="center">New Products Page - Subject: ARTS</H2>
<TABLE ALIGN="CENTER" BORDER="0" WIDTH="660">
<TR ALIGN="CENTER" VALIGN="top">
<TD COLSPAN="5"><B><FONT COLOR="#ff0000" SIZE="+1">Click on one of our latest
books to find out more !</FONT></B></TD></TR>
<TR ALIGN="CENTER" VALIGN="top">
<TD><A HREF="tpcw.dll?BookInfo=59227&Session=6250&C_ID=0"><IMG
SRC="5kgif1.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=24568&Session=6250&C_ID=0"><IMG
SRC="5kgif2.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=94429&Session=6250&C_ID=0"><IMG
SRC="5kgif3.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=68639&Session=6250&C_ID=0"><IMG
SRC="5kgif4.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=49145&Session=6250&C_ID=0"><IMG
SRC="5kgif5.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD></TR></TABLE>
<TABLE BORDER="1" CELLPADDING="1" CELLSPACING="1">
<TR>
<TD WIDTH="30"></TD>
<TD><FONT SIZE="+1">Author</FONT></TD>
<TD><FONT SIZE="+1">Title</FONT></TD></TR>
<TR>
<TD>1.</TD>
<TD><I>stU6Plb4ZT6Grli ViBARESEALNGMF6D9OnI</I></TD>
<TD><A HREF="tpcw.dll?BookInfo=62270&Session=2880021&C_ID=2880021">1OGRENGSERIgVR
            </A></TD></TR>
<TR>
<TD>2.</TD>
<TD><I>EOnhFBqEk1O BANGSEOGBA</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=93648&Session=2880021&C_ID=2880021">KrsfIHwvioZuZOGINOGALSE
M4WiQDCl
            </A></TD></TR>
<TR>
<TD>3.</TD>
<TD><I>ZnPyLyHaMt axk4BARIOGBABAMBJMo3</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=77628&Session=2880021&C_ID=2880021">WgAbL4kp5i2RObmj1WtqXKq
8OGULULREBAComTAXgHnNqenOpezH
            </A></TD></TR>
<TR>
<TD>4.</TD>
<TD><I>dRUZWCGd tnpBAOGULOGRIUE</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=13010&Session=2880021&C_ID=2880021">axdxCeeRrJViu6Y0Xrr0jTq
8BOGRIBAOGBAe1rAkUrk2RfeftySIE
            </A></TD></TR>
<TR>
<TD>5.</TD>
<TD><I>zv7n nBAREATINULBxtrfQ</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=18519&Session=2880021&C_ID=2880021">1I2uPOGINSEOGNGHQw3gXZv
cEAO07UmZzjAbRcCprGJkeBLX
            </A></TD></TR>
<TR>
```

```
<TD>6.</TD>
<TD><I>Jcln seCBARIRIRIRESxF</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=45892&Session=2880021&C_ID=2880021">MREG0dXuqTnBAREATALOGyS
i7Nn
            </A></TD></TR>
<TR>
<TD>7.</TD>
<TD><I>kWYXA BAINRIRERI</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=72948&Session=2880021&C_ID=2880021">GqvBASENGBAATF7xED7GTev
CYGhT
            </A></TD></TR>
<TR>
<TD>8.</TD>
<TD><I>dGFbrl9vKsoJgKkL CBABAALNGBA0WkKll</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=12799&Session=2880021&C_ID=2880021">GnYuA8tcUOGALULNGNGZo
            </A></TD></TR>
<TR>
<TD>9.</TD>
<TD><I>VjE BARENGNGOGVQNzdz</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=907&Session=2880021&C_ID=2880021">1f5oBABANGBAULV9BlrsGlIM5
NCRAoCjbPpUtLe3
            </A></TD></TR>
<TR>
<TD>10.</TD>
<TD><I>VoxuPHKkTgKq2Z1crUh iEoBANGBASEOG5</I></TD>
<TD><A HREF="tpcw.dll?BookInfo=5568&Session=2880021&C_ID=2880021">vqDABASESEATIN
            </A></TD></TR>
<TR>
<TD>11.</TD>
<TD><I>5Iz08WI0ipiwQCLAcCyU m9eAMBARISENGSEK1j8h</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=9432&Session=2880021&C_ID=2880021">IvBANGRERIALkYwPYD60aiNG
zG2oTVErrcrOXv
            </A></TD></TR>
<TR>
<TD>12.</TD>
<TD><I>9mWi18LfLSFmwd SNGX3BANGSEREAL5OiL</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=10486&Session=2880021&C_ID=2880021">8LztbEPf1FOGBAREINATlln
kpVXU2SN4ITpD9114
            </A></TD></TR>
<TR>
<TD>13.</TD>
<TD><I>in8UJrIdDeU srkgrBAULSEALRI1E</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=40122&Session=2880021&C_ID=2880021">1h2qOGATULININfdCId8
            </A></TD></TR>
<TR>
<TD>14.</TD>
<TD><I>J8jn3qhul8q4 BAREINRENG</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=97797&Session=2880021&C_ID=2880021">4BAREALALNGbpCuOtEOvmuf
CMFzDQcFfD66oCUxjC
            </A></TD></TR>
<TR>
<TD>15.</TD>
<TD><I>DPNYuaOo4IHeneTbA 3PylocCXb2BAULATALSE</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=39900&Session=2880021&C_ID=2880021">wxOHSyU6NMrDWT46Z9dOGAT
OGNGUL1a08NvSrPO
            </A></TD></TR>
<TR>
<TD>16.</TD>
<TD><I>JiiIWu11dOZBIyHf8l c8BAULRIINNG</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=3711&Session=2880021&C_ID=2880021">4c1xwNdVA5BARIULOGOGXR0k
9GBbXqfVNIzT0lf
            </A></TD></TR>
<TR>
```

```
<TD>17.</TD>
<TD><I>RUaDF zBAATULOGRIn</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=64989&Session=2880021&C_ID=2880021">0B4bqdBkLK5bvOGATRIBANG
ZsCtzGl3iNAye
            </A></TD></TR>
<TR>
<TD>18.</TD>
<TD><I>3GIjs hBAULOGATOGJx</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=55618&Session=2880021&C_ID=2880021">DxAq1Ud0t5FEBANGRIRESEW
snJX8duztXxhooHDIxzRUjGeuJG
            </A></TD></TR>
<TR>
<TD>19.</TD>
<TD><I>yF01NUWh7QdbOEq q5XNBAULRIBAULh</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=87316&Session=2880021&C_ID=2880021">Rj5adv0IRzFxQBAULULULUL
0jw1VViD2KxCnsl5vKjta9srh
            </A></TD></TR>
<TR>
<TD>20.</TD>
<TD><I>Bbr9D e0z9vdvhahBABARIALNG</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=52201&Session=2880021&C_ID=2880021">JJiRMegCqE6vXJfM3Y4PBAA
TSEREINeFmjSW
            </A></TD></TR>
<TR>
<TD>21.</TD>
<TD><I>VdI2K0HVn 2XvBABAATATAL</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=35106&Session=2880021&C_ID=2880021">rJwVRHHC78MxwUjokDy88fs
EFajGlrRzUOGRERIALREFopJXG
            </A></TD></TR>
<TR>
<TD>22.</TD>
<TD><I>HLUYa4nDmovkVTYUxN rZuBAATATINNGDSBcC</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=5870&Session=2880021&C_ID=2880021">PU1QOc9BzBASEINULBAcqk3
            </A></TD></TR>
<TR>
<TD>23.</TD>
<TD><I>8RoDS2Z0WbYkF BARIRIRERI</I></TD>
<TD><A HREF="tpcw.dll?BookInfo=8572&Session=2880021&C_ID=2880021">LrBAINSEULALgouHP
            </A></TD></TR>
<TR>
<TD>24.</TD>
<TD><I>hzDIq5z4dRbFos5Mx38 Nq8BAULNGBARE</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=12772&Session=2880021&C_ID=2880021">InxLMioT7mZOGORcrY6O6Lm
a1nKysmOGALULULAL
            </A></TD></TR>
<TR>
<TD>25.</TD>
<TD><I>jQ1tbivDFAnVYBm44 BAOGOGNGREv</I></TD>
<TD><A HREF="tpcw.dll?BookInfo=51256&Session=2880021&C_ID=2880021">QCbPKBAATOGALRE
            </A></TD></TR>
<TR>
<TD>26.</TD>
<TD><I>lY7uBmqfsAywXKdTSV BAALINOGRE</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=30411&Session=2880021&C_ID=2880021">wDhy761Yptsctbeyg9pstXK
aBoxBARIATRINGDis5E2ijT7oUkE3jFCM
            </A></TD></TR>
<TR>
<TD>27.</TD>
<TD><I>9aWJ74y7CxPB71rIr3 BARENGRENGEdjn</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=82215&Session=2880021&C_ID=2880021">t0BsEOGSERIALNGHqGR2hQ5
SQMWrI
            </A></TD></TR>
<TR>
<TD>28.</TD>
```

```
<TD><I>4zHnYGWjZsYQRPhr HBABAINSESEhQpxFjb</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=46592&Session=2880021&C_ID=2880021">84dNJ9L1BABABAOGULtqeDJ
Vcr6VSAdXCX
            </A></TD></TR>
<TR>
<TD>29.</TD>
<TD><I>w4IZupIi7o98rTQ2 gqKA2RFIBAOGSEBABAFU</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=48569&Session=2880021&C_ID=2880021">N7AOGOGRISESECfBX9rgLFA
YzHUO1O4W5FVV1WLFw8K
            </A></TD></TR>
<TR>
<TD>30.</TD>
<TD><I>dzhXQntc7WaVC BASEULULIN</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=84540&Session=2880021&C_ID=2880021">d9JSJBCqYYNnBtr3x26BBAA
TRIRIATJIpQzm4CslKw44z13PeC
            </A></TD></TR>
<TR>
<TD>31.</TD>
<TD><I>elS0PutEc340RR1FK ZrBASEULSEUL</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=43017&Session=2880021&C_ID=2880021">SaFjn8mjLjBvYOGSEREALSE
EDk4p1dU
            </A></TD></TR>
<TR>
<TD>32.</TD>
<TD><I>hFd0qwdo BAULBAINAT</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=45034&Session=2880021&C_ID=2880021">eWD5cfJHZh2HDneIOGALREB
AINW0Xnn2G4CKWJ6n2JS6fwsNc
            </A></TD></TR>
<TR>
<TD>33.</TD>
<TD><I>USa0zHZ2qEzAshTM Jb3kDBAINBAATULL</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=48288&Session=2880021&C_ID=2880021">2SSUi8BQznlzVb9mtQc2kfz
Iy9hE3lW5WBARIRERERIYHj13Htfct
            </A></TD></TR>
<TR>
<TD>34.</TD>
<TD><I>eyubq6Ecs7 ie1BAOGNGRIREOXLNlCC</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=15689&Session=2880021&C_ID=2880021">ykI0C5pmAorOGSEATINNGVa
nCzJQgHpHl5xeuPXF8wBXuV1YKS
            </A></TD></TR>
<TR>
<TD>35.</TD>
<TD><I>IaBhg4w Z7BARIINRERIlm4aeF6</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=64431&Session=2880021&C_ID=2880021">6OGOGBAATUL0MOVhMW5k0Z3
8eDNMuDHywZvKG7Mzhalx6xiZM4SPB9
            </A></TD></TR>
<TR>
<TD>36.</TD>
<TD><I>0T1hbcP BAALSEULOGcKN</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=80340&Session=2880021&C_ID=2880021">FPTg2a9q2x1RKhrlQZoGOOG
BAREINBApG
            </A></TD></TR>
<TR>
<TD>37.</TD>
<TD><I>x23ZPje2ANQkpiTQ yjRh17EBABAALBABA</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=97861&Session=2880021&C_ID=2880021">BR2kTOU3iZ1SI32r9Na4J2r
9essqZsHSOGNGBABANGYslmkB5LoIAnqByQZP
            </A></TD></TR>
<TR>
<TD>38.</TD>
<TD><I>hkPT MYo6RNBAREOGATOGWU</I></TD>
```

```
<TD><A
HREF="tpcw.dll?BookInfo=88976&Session=2880021&C_ID=2880021">wG5QOiK7PGXQPkBABANGRIS
E2F
            </A></TD></TR>
<TR>
<TD>39.</TD>
<TD><I>gpC8UE BAALNGBAATo</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=48881&Session=2880021&C_ID=2880021">8Rr5OGINRISEOGgyj
            </A></TD></TR>
<TR>
<TD>40.</TD>
<TD><I>uOjUkg311fFVHNY c1ymBAINULBABA3H5</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=50141&Session=2880021&C_ID=2880021">i8pR3xCOGSERIBAULu7
            </A></TD></TR>
<TR>
<TD>41.</TD>
<TD><I>YKJ7EjFTepimtcKAQuD fEceBARIOGINNGJr6</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=48561&Session=2880021&C_ID=2880021">CHPHSMbAHOGBARIATAL3An
            </A></TD></TR>
<TR>
<TD>42.</TD>
<TD><I>WydV6nf wZQBAALSEATBAI3</I></TD>
<TD><A HREF="tpcw.dll?BookInfo=6275&Session=2880021&C_ID=2880021">BAATALULSE5VvYkV
            </A></TD></TR>
<TR>
<TD>43.</TD>
<TD><I>AiVg5L5xoIYns4duE lHjkhMGWBABAULNGINLC</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=49074&Session=2880021&C_ID=2880021">tJ915cy3XXiw5GTiH8Kexkd
xACCoB6kGepMuupnOGATULRISEX7u
            </A></TD></TR>
<TR>
<TD>44.</TD>
<TD><I>bD8fP6n1t 6AQcBAULRIRIAT6T</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=74983&Session=2880021&C_ID=2880021">y47S6pbnFjNcnOGNGREINUL
omIqoy
            </A></TD></TR>
<TR>
<TD>45.</TD>
<TD><I>XRtBAGg1Ppai41Z95L jBAALSEULATU</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=25222&Session=2880021&C_ID=2880021">Poxk6UDuOXUXBAINBAULATX
5jciYRwEjLra6ciLVvxZUn7yP
            </A></TD></TR>
<TR>
<TD>46.</TD>
<TD><I>W8S22 BARESEATRI</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=91728&Session=2880021&C_ID=2880021">eXtFR1gZnOGINBARISEbHOB
yUn
            </A></TD></TR>
<TR>
<TD>47.</TD>
<TD><I>9aoVyxHNyQ 4mBAREOGSEIN6W</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=15966&Session=2880021&C_ID=2880021">zsit9eiWOGSENGATATTwDHv
BM72JaQ0DZAbhomd2Mk6vEWi
            </A></TD></TR>
<TR>
<TD>48.</TD>
<TD><I>W3T3e7zGBr1mUQw 8mBARIBARIRIxERPcW</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=23270&Session=2880021&C_ID=2880021">tpf7WVEROGNGALREOG9jpuw
r79LYceHk7KhUxjI2VR
            </A></TD></TR>
<TR>
<TD>49.</TD>
<TD><I>HqVmdTJ9YBr d7N3h4vBAINALRIALQs</I></TD>
```

```
<TD><A
HREF="tpcw.dll?BookInfo=85519&Session=2880021&C_ID=2880021">ItMOGOGULOGULuCDFQceP4e
20
             </A></TD></TR>
<TR>
<TD>50.</TD>
<TD><I>1O0t6h ij9d6FFBAREREBAALZi</I></TD>
<TD><A
HREF="tpcw.dll?BookInfo=4723&Session=2880021&C_ID=2880021">E34SctIMqrFdID1L4SBAREUL
ALRIcbHOaAoMvUB
             </A></TD></TR></TABLE>
<P ALIGN="center"><A
HREF="tpcw.dll?ShoppingBasket&ADD_FLAG=N&Session=2880021&C_ID=2880021"><IMG
SRC="cart.gif" ALT="Shopping Cart" BORDER="0" WIDTH="120" HEIGHT="30"></A><A
HREF="tpcw.dll?search&Session=2880021&C_ID=2880021"><IMG
SRC="search.gif" ALT="Search" BORDER="0" WIDTH="120" HEIGHT="30"></A><A
HREF="tpcw.dll?home&Session=2880021&C_ID=2880021"><IMG
SRC="home.gif" ALT="Home" BORDER="0" WIDTH="120" HEIGHT="30"></A></P></BODY></HTML>
```

F.9 Order Display

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN">
<HTML>
<HEAD>
<TITLE>TPC-W Order Display Page</TITLE></HEAD>
<BODY BGCOLOR="#FFFFFF">
<H1 ALIGN="CENTER">TPC Web Commerce Benchmark (TPC-W)</H1>
<H2 ALIGN="CENTER">Order Display Page</H2>
<BLOCKQUOTE>
<BLOCKQUOTE>
<BLOCKQUOTE>
<BLOCKQUOTE>
<HR>
<P>Order ID: 2592013<BR>Order Placed on 4/1/1999<BR>Shipping Type: Regular<BR>Ship
Date: 808464674/141402<BR>Order Subtotal: $968.97<BR>Order Tax: $79.94<BR>Order
Total: $1054.91<BR></P>
<TABLE BORDER="0" WIDTH="80%">
<TR>
<TD><B>Bill To:</B></TD>
<TD><B>Ship To:</B></TD></TR>
<TR>
<TD COLSPAN="2">
<H4>John Doe</H4></TD></TR>
<TR>
<TD WIDTH="50%"><ADDRESS>1 Some Place<BR>Appt 1<BR>Dullsville<BR>CA
92691<BR>Libya<BR><BR>Email: me@work.com<BR>Phone: 123-456-7890</ADDRESS><BR>
<P>Credit Card Type: Visa<BR>Order Status: Pending</P></TD>
<TD VALIGN="TOP" WIDTH="50%"><ADDRESS>1 Some Place<BR>Appt 1<BR>Dullsville<BR>CA
92691<BR>Libya</ADDRESS></TD></TR></TABLE></BLOCKQUOTE></BLOCKQUOTE></BLOCKQUOTE></
BLOCKQUOTE>
<CENTER>
<TABLE BORDER="1" CELLPADDING="5" CELLSPACING="0">
<TR>
<TD>
<H4>Item #</H4></TD>
<TD>
<H4>Title</H4></TD>
<TD>
<H4>Cost</H4></TD>
<TD>
<H4>Qty</H4></TD>
<TD>
<H4>Discount</H4></TD>
<TD>
<H4>Comment</H4></TD></TR>
<TR>
<TD>
<H4>5314</H4></TD>
<TD VALIGN="top">
<H4>Title: Xjg1CdU73Q7KQRZ0fd7BASERIOGREmY1JHeT11op0z9m<BR>Publisher:
```

```
9XVWJDR8WGOEps4</H4></TD>
<TD>
<H4>$930.23</H4></TD>
<TD>
<H4>1</H4></TD>
<TD>
<H4>1.00</H4></TD>
<TD>
<H4>comment</H4></TD></TR>
<TR>
<TD>
<H4>62440</H4></TD>
<TD VALIGN="top">
<H4>Title: ojPavZt5Co6LW57SEKy7BB22ODy8OGSEALBABA5isiu<BR>Publisher:
3dGH107y2oxeGOqWxYMajVTk8zw8hZdH7</H4></TD>
<TD>
<H4>$658.25</H4></TD>
<TD>
<H4>1</H4></TD>
<TD>
<H4>1.00</H4></TD>
<TD>
<H4>comment</H4></TD></TR></TABLE><BR><A
HREF="tpcw.dll?search&Session=0&C_ID=0"><IMG
SRC="search.gif" ALT="Search" BORDER="0" WIDTH="120" HEIGHT="30"></A>
<A HREF="tpcw.dll?home&Session=0&C_ID=0"><IMG
SRC="home.gif" ALT="Home" BORDER="0" WIDTH="120"
HEIGHT="30"></A></CENTER></BODY></HTML>
```

F.10 Order Inquiry

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN">
<HTML>
<HEAD>
<TITLE>TPC-W Order Inquiry Page</TITLE></HEAD>
<BODY BGCOLOR="#ffffff">
<H1 ALIGN="center">TPC Web Commerce Benchmark (TPC-W)</H1>
<H2 ALIGN="center">Order Inquiry Page</H2>
<FORM ACTION="tpcw.dll?OrderResult&amp;Session=0&amp;C_ID=0" METHOD="post">
<TABLE ALIGN="CENTER">
<TR>
<TD>
<H4>Username:</H4></TD>
<TD>            <INPUT NAME="username" SIZE="23"></TD></TR>
<TR>
<TD>
<H4>Password:</H4></TD>
<TD>
<INPUT NAME="password" SIZE="14" TYPE="password"
VALUE="password"></TD></TR></TABLE>
<P ALIGN="CENTER"><BR><INPUT
TYPE="IMAGE" NAME="Display Last Order" SRC="ord_sta.gif" HEIGHT="30" WIDTH="120">
<A HREF="tpcw.dll?search&Session=0&C_ID=0"><IMG
SRC="search.gif" ALT="Search" BORDER="0" WIDTH="120" HEIGHT="30"></A>
<A HREF="tpcw.dll?home&Session=0&C_ID=0"><IMG
SRC="home.gif" ALT="Home" BORDER="0" WIDTH="120"
HEIGHT="30"></A></P></FORM></BODY></HTML>
```

F.11 Product Detail

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN">
<HTML>
<HEAD>
<TITLE>TPC-W Product Detail Page</TITLE></HEAD>
<BODY BGCOLOR="#ffffff">
<H1 ALIGN="center">TPC Web Commerce Benchmark (TPC-W)</H1>
<CENTER><IMG
SRC="tpclogo.gif" ALIGN="BOTTOM" BORDER="0" WIDTH="288" HEIGHT="67"></CENTER>
```

```
<H2 ALIGN="center">Product Detail Page</H2>
<H2>Title: 1OGRENGSERIgVR</H2>
<P>Author: stU6Plb4ZT6Grli h0WTS<BR>Subject: ARTS</P>
<P><IMG SRC="25k.gif" ALIGN="RIGHT" BORDER="0" WIDTH="200" HEIGHT="200">
Description: <I>zegEZtxoZQFHY6YqvFRt79tfdUno6fFZz0Jb5t7W93JrBubUWfz4FjVo7S4Nt
OAQMXJ3kHD8kbKvU0SB0E2SUkGlDJ0653ZFYaoCmE  ZbPMSL4w3GLOWxFCmYaEQqP</I></P>
<BLOCKQUOTE>
<P><B>Suggested Retail: $986.25</B><BR><B>Our Price:</B><FONT
COLOR="#dd0000"><B> $269.83</B></FONT><BR><B>You Save:</B><FONT
COLOR="#dd0000"><B> $716.42</B></FONT></P></BLOCKQUOTE>
<DL>
<DT><FONT SIZE="2">Backing: HARDBACK, 12364 pages<BR>Published by
rxaskEE1HACDzlz7a9WbzcNJ1nv3Yt6VdRM5NvUg<BR>Publication date:      12-30-
1999<BR>Avail
date:      01-04-2000<BR>Dimensions (in inches): 29.37 x 38.59 x 55.70<BR>ISBN:
     fghxUVgInPPIw</FONT></DT></DL>
<P ALIGN="center"><A
HREF="tpcw.dll?ShoppingBasket&ADD_FLAG=Y&BookID=62270&Title=1OGRENGSERIgVR&SRP=986.
25&Cost=269.83&Packing=HARDBACK&Session=6251&C_ID=0"><IMG
SRC="add.gif" ALT="Add to Basket" BORDER="0" WIDTH="120" HEIGHT="30"></A><A
HREF="tpcw.dll?search&Session=6251&C_ID=0"><IMG
SRC="search.gif" ALT="Search" BORDER="0" WIDTH="120" HEIGHT="30"></A><A
HREF="tpcw.dll?home&Session=6251&C_ID=0"><IMG
SRC="home.gif" ALT="Home" BORDER="0" WIDTH="120" HEIGHT="30"></A><A
HREF="tpcw.dll?ShowBook&BookID=62270&Title=1OGRENGSERIgVR&A_Fname=stU6Plb4ZT6Grli&A
_Lname=h0WTS&ImagePath=70&Image=i62270.gif&SRP=986.25&Cost=269.83&Session=6251&C_ID
=0"><IMG
SRC="update.gif" ALT="Update" BORDER="0" WIDTH="120"
HEIGHT="30"></A></P></BODY></HTML>
```

F.12 Search Request

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN">
<HTML>
<HEAD>
<TITLE>Search Request Page</TITLE></HEAD>
<BODY>
<CENTER>
<H1 ALIGN="center">TPC W Commerce Benchmark (TPC-W)</H1>
<H2 ALIGN="center"><IMG
SRC="tpclogo.gif" ALIGN="BOTTOM" BORDER="0" WIDTH="288" HEIGHT="67"></H2>
<H2 ALIGN="center">Search Request Page</H2>
<TABLE ALIGN="CENTER" BORDER="0" WIDTH="660">
<TR ALIGN="CENTER" VALIGN="top">
<TD COLSPAN="5"><B><FONT COLOR="#ff0000" SIZE="+1">Click on one of our latest
books to find out more !</FONT></B></TD></TR>
<TR ALIGN="CENTER" VALIGN="top">
<TD><A HREF="tpcw.dll?BookInfo=59227&Session=6250&C_ID=0"><IMG
SRC="5kgif1.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=24568&Session=6250&C_ID=0"><IMG
SRC="5kgif2.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=94429&Session=6250&C_ID=0"><IMG
SRC="5kgif3.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=68639&Session=6250&C_ID=0"><IMG
SRC="5kgif4.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=49145&Session=6250&C_ID=0"><IMG
SRC="5kgif5.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD></TR></TABLE>
</CENTER>
<FORM ACTION="tpcw.dll?result&amp;Session=5678&amp;C_ID=2880005" METHOD="post">
<TABLE ALIGN="center">
<TR>
<TD ALIGN="right">
<H3>Search by:</H3></TD>
<TD WIDTH="100"></TD></TR>
<TR>
<TD ALIGN="right">
<SELECT NAME="search" SIZE="1">
<OPTION SELECTED="SELECTED" VALUE="author">Author                    </OPTION>
<OPTION VALUE="title">Title                    </OPTION>
<OPTION VALUE="subject">Subject</OPTION></SELECT></TD>
```

```
<TD>            <INPUT NAME="title_field" SIZE="30"></TD></TR></TABLE>
<P ALIGN="center"><INPUT
TYPE="IMAGE" NAME="Search" SRC="submit.gif" HEIGHT="30" WIDTH="120"><A
HREF="tpcw.dll?home&Session=5678&C_ID=2880005"><IMG
SRC="home.gif" ALT="Home" BORDER="0" WIDTH="120" HEIGHT="30"></A><A
HREF="tpcw.dll?ShoppingBasket&ADD_FLAG=N&Session=5678&C_ID=2880005"><IMG
SRC="cart.gif" ALT="Shopping Cart" BORDER="0" WIDTH="120" HEIGHT="30"></A>
</P></FORM></BODY></HTML>
```

F.13 Search Results

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN">
<HTML>
<HEAD>
<TITLE>Search Result</TITLE></HEAD>
<BODY>
<H1 ALIGN="CENTER">TPC Web Commerce Benchmark (TPC-W)</H1>
<H2 ALIGN="CENTER"><IMG
SRC="tpclogo.gif" ALIGN="BOTTOM" BORDER="0" WIDTH="288" HEIGHT="67"></H2>
<H2 ALIGN="CENTER">Search Result Page - Title: OGALRIRESE</H2>
<TABLE ALIGN="CENTER" BORDER="0" WIDTH="660">
<TR ALIGN="CENTER" VALIGN="top">
<TD COLSPAN="5"><B><FONT COLOR="#ff0000" SIZE="+1">Click on one of our latest
books to find out more !</FONT></B></TD></TR>
<TR ALIGN="CENTER" VALIGN="top">
<TD><A HREF="tpcw.dll?BookInfo=59227&Session=6250&C_ID=0"><IMG
SRC="5kgif1.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=24568&Session=6250&C_ID=0"><IMG
SRC="5kgif2.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=94429&Session=6250&C_ID=0"><IMG
SRC="5kgif3.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=68639&Session=6250&C_ID=0"><IMG
SRC="5kgif4.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=49145&Session=6250&C_ID=0"><IMG
SRC="5kgif5.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD></TR></TABLE>
<TABLE BORDER="1" CELLPADDING="1" CELLSPACING="1">
<TR>
<TD WIDTH="30"></TD>
<TD><FONT SIZE="+1">Author</FONT></TD>
<TD><FONT SIZE="+1">Title</FONT></TD></TR>
<TR>
<TD>1.</TD>
<TD><I>gXvHNxEO7t1M8NWSGWo 5nSSBAALBANGALh</I></TD>
<TD><A HREF="tpcw.dll?BookInfo=12345&Session=5680&C_ID=0">
m7OGALRIRESEv34tVPXPM0ttGeD2IBm0w8 </A></TD></TR>
<TR>
<TD>2.</TD>
<TD><I>0MbULYSFsG 5bBAOGINREULcQoHw6</I></TD>
<TD><A HREF="tpcw.dll?BookInfo=40984&Session=5680&C_ID=0">
nvttzGLAp0scOGALRIRESEqDSI97TV0LHSvUNjG1S82N7buJ </A></TD></TR></TABLE>
<P ALIGN="CENTER"><A
HREF="tpcw.dll?ShoppingBasket&ADD_FLAG=N&Session=5680&C_ID=0"><IMG
SRC="cart.gif" ALT="Shopping Cart" BORDER="0" WIDTH="120" HEIGHT="30"></A><A
HREF="tpcw.dll?search&Session=5680&C_ID=0"><IMG
SRC="search.gif" ALT="Search" BORDER="0" WIDTH="120" HEIGHT="30"></A><A
HREF="tpcw.dll?home&Session=5680&C_ID=0"><IMG
SRC="home.gif" ALT="Home" BORDER="0" WIDTH="120" HEIGHT="30"></A></P></BODY></HTML>
```

F.14 Shopping Cart

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD W3 HTML//EN">
<HTML>
<HEAD>
<TITLE>TPC W Shopping Cart</TITLE></HEAD>
<BODY BGCOLOR="#ffffff">
<H1 ALIGN="center">TPC Web Commerce Benchmark (TPC-W)</H1>
<CENTER><IMG
SRC="tpclogo.gif" ALIGN="BOTTOM" BORDER="0" WIDTH="288" HEIGHT="67"></CENTER>
```

```
<H2 ALIGN="center">Shopping Cart Page</H2>
<TABLE ALIGN="CENTER" BORDER="0" WIDTH="660">
<TR ALIGN="CENTER" VALIGN="top">
<TD COLSPAN="5"><B><FONT COLOR="#ff0000" SIZE="+1">Click on one of our latest
books to find out more !</FONT></B></TD></TR>
<TR ALIGN="CENTER" VALIGN="top">
<TD><A HREF="tpcw.dll?BookInfo=59227&Session=6250&C_ID=0"><IMG
SRC="5kgif1.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=24568&Session=6250&C_ID=0"><IMG
SRC="5kgif2.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=94429&Session=6250&C_ID=0"><IMG
SRC="5kgif3.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=68639&Session=6250&C_ID=0"><IMG
SRC="5kgif4.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD>
<TD><A HREF="tpcw.dll?BookInfo=49145&Session=6250&C_ID=0"><IMG
SRC="5kgif5.gif" ALT="Book 1" WIDTH="100" HEIGHT="150"></A></TD></TR></TABLE>
<FORM ACTION="tpcw.dll?Adjustment&amp;Session=5678&amp;C_ID=0" METHOD="post">
<CENTER>
<P></P>
<TABLE BORDER="0">
<TR>
<TD><B>Qty</B></TD>
<TD><B>Product</B></TD></TR>
<TR>
<TD VALIGN="top">            <INPUT NAME="QTY" SIZE="3" VALUE="1"></TD>
<TD VALIGN="top">Title:<I>1OGRENGSERIgVR</I> - Backing:HARDBACK<BR>SRP.
$986.25,              <FONT COLOR="#aa0000"><B>Your Price:
$269.83</B></FONT></TD></TR>
<TR>
<TD VALIGN="top">            <INPUT NAME="QTY" SIZE="3" VALUE="1"></TD>
<TD VALIGN="top">Title:<I>KrsfIHwvioZuZOGINOGALSEM4WiQDCl</I> -
Backing:HARDBACK <BR>SRP. $701.72,
<FONT COLOR="#aa0000"><B>Your Price:              $500.17</B></FONT></TD></TR>
<TR>
<TD VALIGN="top">            <INPUT NAME="QTY" SIZE="3" VALUE="2"></TD>
<TD VALIGN="top">Title:<I>axdxCeeRrJViu6Y0Xrr0jTq8BOGRIBAOGBAe1rAkUrk2RfeftySIE</I>
             - Backing:AUDIO <BR>SRP. $270.11,
<FONT COLOR="#aa0000"><B>Your Price:
$257.18</B></FONT></TD></TR></TABLE><B><I>Subtotal
price: $1284.36
</I></B>
<P><BR><INPUT
TYPE="IMAGE" NAME="Proceed to Checkout" SRC="checkout.gif" HEIGHT="30"
WIDTH="120"><INPUT
TYPE="IMAGE" NAME="Continue Shopping" SRC="home.gif" HEIGHT="30" WIDTH="120"></P>
<P>If you have changed the quantities and/or taken anything out<BR>of your
shopping cart, click here to refreh your shopping cart:</P>
<P><INPUT
TYPE="IMAGE" NAME="Refresh Shopping Cart" SRC="refresh.gif" HEIGHT="30"
WIDTH="120"></P></CENTER></FORM></BODY></HTML>
```

Appendix G -- Reporting Templates

F.1 Executive summary

[TO BE COMPLETED]