US007548867B2

(12) **United States Patent**
Halavais et al.

(10) **Patent No.: US 7,548,867 B2**
(45) **Date of Patent: Jun. 16, 2009**

(54) **SYSTEM AND METHOD FOR MAINTAINING COHERENCY OF DATA ENTRIES**

(75) Inventors: **Richard Arthur Halavais**, Anaheim Hills, CA (US); **Tony Cheng-Tong Chung**, Diamond Bar, CA (US)

(73) Assignee: **Ceats, Inc.**, Las Vegas, NV (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **12/329,512**

(22) Filed: **Dec. 5, 2008**

(65) **Prior Publication Data**

US 2009/0083083 A1 Mar. 26, 2009

Related U.S. Application Data

(63) Continuation of application No. 12/244,727, filed on Oct. 2, 2008, which is a continuation of application No. 09/295,577, filed on Apr. 22, 1999, now Pat. No. 7,454,361.

(51) **Int. Cl.**
G06Q 10/00 (2006.01)

(52) **U.S. Cl.** **705/5**

(58) **Field of Classification Search** 705/1, 705/5

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,427,438	A	2/1969	Ruscher
4,449,186	A	5/1984	Kelly et al.
4,794,530	A	12/1988	Yukiura et al.
4,931,932	A	6/1990	Dalnekoff et al.
4,974,252	A	11/1990	Osborne
4,984,156	A	1/1991	Mekata

(Continued)

FOREIGN PATENT DOCUMENTS

GB	2366403	A	8/2000

(Continued)

OTHER PUBLICATIONS

Anderson, Jennifer, "Ticket to Ride", Apr. 21, 1998, PC Magazine, v17, n8, p. 50 (1).*

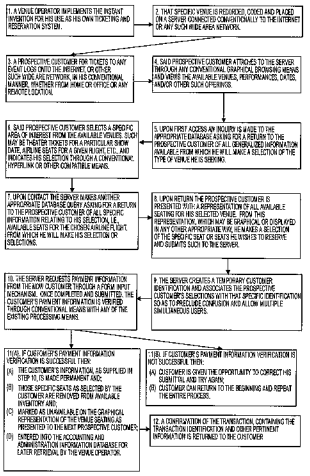
(Continued)

Primary Examiner—John W Hayes
Assistant Examiner—Shannon S Saliard
(74) *Attorney, Agent, or Firm*—Knobbe Martens Olson & Bear LLP

(57) **ABSTRACT**

The present invention relates to a system and method for maintaining coherency of data entries related to seat information obtained from remote venues. The method handles queries regarding seat status, including information indicating seat availability. Information obtained based on the queries are transmitted to a first general purpose computer for display to a first user. Upon receiving, from the first general purpose computer, selections by the first user, the system or method creates in the database a temporary record based on the selections and requests a confirmation of the selections by the first user. Once the first user sends the confirmation, the system or method locks entries in the database, preventing a second user from selecting any seat confirmed for the first user. The system or method also includes transmitting, to a second general purpose computer, information to be displayed to the second user indicating seats selected by the first.

9 Claims, 30 Drawing Sheets



U.S. PATENT DOCUMENTS

4,992,647	A		2/1991	Konishi et al.
5,021,953	A		6/1991	Webber et al.
5,027,224	A		6/1991	Yamada
5,151,692	A		9/1992	Hirahara
5,237,499	A		8/1993	Garback
5,239,480	A		8/1993	Huegel
5,255,184	A		10/1993	Hornick et al.
5,270,921	A		12/1993	Hornick et al.
5,285,055	A		2/1994	Oonakahara et al.
5,311,425	A		5/1994	Inada
5,333,257	A		7/1994	Merrill et al.
5,408,417	A		4/1995	Wilder
5,581,461	A		12/1996	Coll et al.
5,615,342	A		3/1997	Johnson
5,621,430	A		4/1997	Bricklin
5,668,592	A		9/1997	Spaulding, II et al.
5,724,520	A		3/1998	Goheen
5,732,398	A		3/1998	Tagawa
5,754,654	A		5/1998	Hiroya et al.
5,769,269	A		6/1998	Peters
5,781,892	A		7/1998	Hunt et al.
5,797,126	A		8/1998	Helbling et al.
5,797,127	A		8/1998	Walker et al.
5,806,046	A		9/1998	Curran et al.
5,808,894	A		9/1998	Wiens et al.
5,832,451	A		11/1998	Flake et al.
5,832,454	A		11/1998	Jafri et al.
5,842,176	A		11/1998	Hunt et al.
5,855,006	A		12/1998	Huemoeller et al.
5,897,620	A		4/1999	Walker et al.
5,926,108	A		7/1999	Wicks et al.
5,926,798	A		7/1999	Carter
5,930,761	A		7/1999	O'Toole
5,930,769	A		7/1999	Rose
5,943,651	A		8/1999	Oosawa
5,948,040	A		9/1999	DeLorme et al.
5,974,406	A		10/1999	Bisdikian et al.
5,990,927	A		11/1999	Hendricks et al.
6,023,685	A		2/2000	Brett et al.
6,070,149	A		5/2000	Tavor et al.
6,085,169	A		7/2000	Walker et al.
6,085,976	A		7/2000	Sehr et al.
6,107,932	A		8/2000	Walker et al.
6,112,185	A		8/2000	Walker et al.
6,119,096	A		9/2000	Mann et al.
6,134,534	A		10/2000	Walker et al.
6,167,462	A	*	12/2000	Davis et al. 710/5
6,223,166	B1		4/2001	Kay
6,240,396	B1		5/2001	Walker et al.
6,249,767	B1		6/2001	Okayama et al.
6,263,315	B1		7/2001	Talluri
6,301,574	B1		10/2001	Thomas et al.
6,304,850	B1		10/2001	Keller et al.
6,380,959	B1		4/2002	Wang et al.
6,418,413	B2		7/2002	DeMarcken et al.
6,606,101	B1		8/2003	Malamud et al.
6,704,713	B1		3/2004	Brett
6,824,057	B2		11/2004	Rathus et al.
6,975,856	B2		12/2005	Ogasawara
6,999,936	B2		2/2006	Sehr
7,010,533	B1		3/2006	Kutsumi et al.
7,069,228	B1		6/2006	Rose et al.
7,114,179	B1		9/2006	Ritter et al.
2001/0016825	A1		8/2001	Pugliese et al.
2001/0043210	A1	*	11/2001	Gilbert et al. 345/420
2002/0156661	A1		10/2002	Jones et al.
2002/0178034	A1		11/2002	Gardner et al.
2004/0107439	A1		6/2004	Hassell et al.
2005/0144115	A1		6/2005	Brett
2005/0199713	A1		9/2005	Rathus et al.
2007/0027794	A1		2/2007	Brett
2007/0027798	A1		2/2007	Brett
2007/0033131	A1		2/2007	Brett
2007/0038582	A1		2/2007	Brett
2007/0124232	A1		5/2007	Brett

FOREIGN PATENT DOCUMENTS

JP	62132491	12/1998

OTHER PUBLICATIONS

"ElectroTix Offers New Visual Approach to Selling Tickets on Internet", May 28, 1998, Business Wire, p052820356.*

Ackerman, *No Ticket No Problem: Ticketless Travel Heralds a Computer-Driven Revolution in the Travel Industry*, Boston Globe, Jun. 4, 1995.

ElectroTix Offers New Visual Approach to Selling Tickets On Internet, May 1998, Business Wire.

Anderson, Jennifer, Ticket to Ride, Apr. 21, 1998, PC Magazine, v17, n8, p. 40(1).

Printout of Travelocity.com Seat Map archived Dec. 12, 1998 from http://web.archive.org/web/19990428171540/www2.travelocity.com/ttools/seatmap.html downloaded Sep. 23, 2008.

Exciting New Features Now Available on Microsoft Expedia, May 6, 1997, PR Newswire.

* cited by examiner

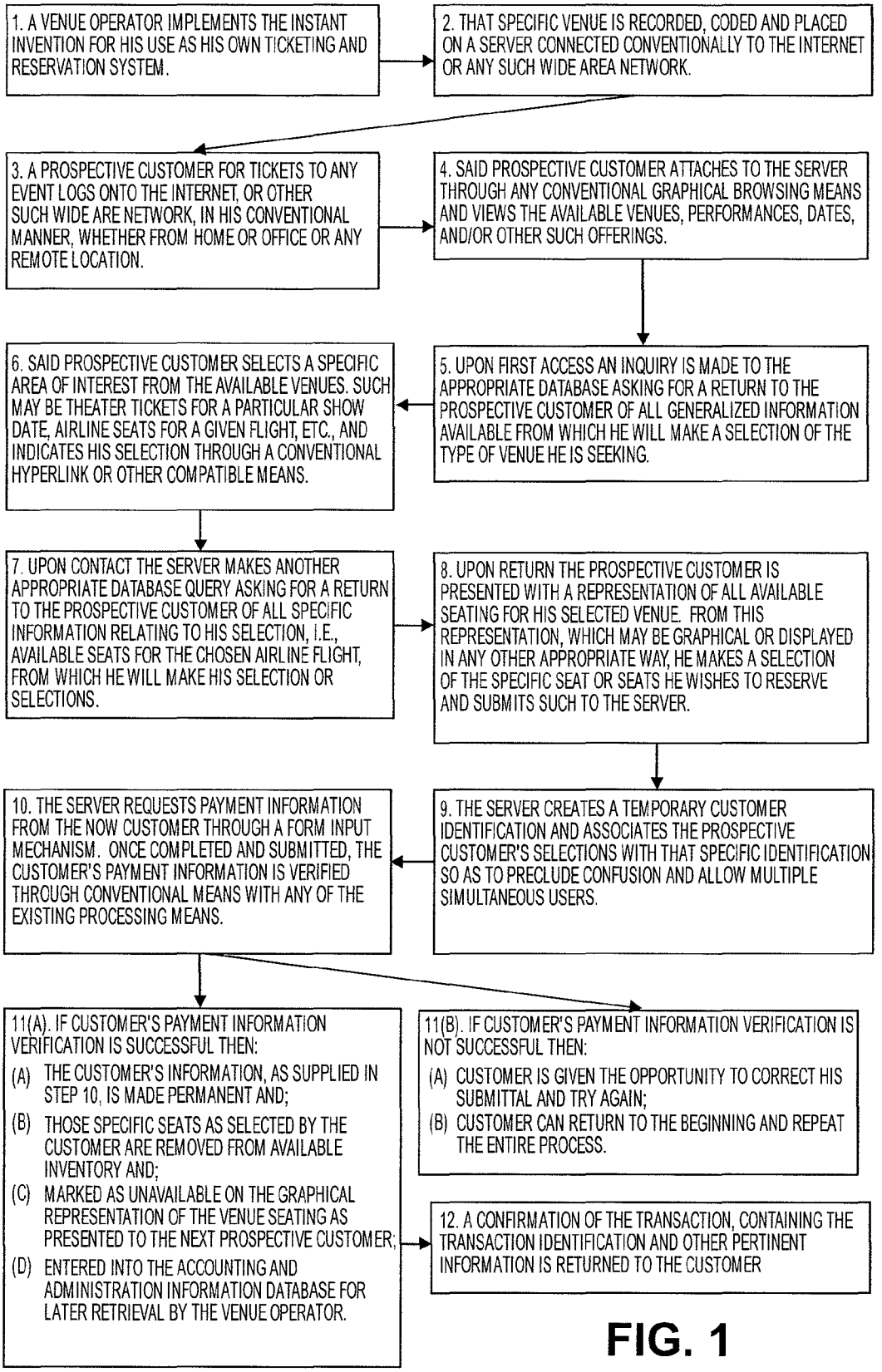

1. A VENUE OPERATOR IMPLEMENTS THE INSTANT INVENTION FOR HIS USE AS HIS OWN TICKETING AND RESERVATION SYSTEM.
2. THAT SPECIFIC VENUE IS RECORDED, CODED AND PLACED ON A SERVER CONNECTED CONVENTIONALLY TO THE INTERNET OR ANY SUCH WIDE AREA NETWORK.
3. A PROSPECTIVE CUSTOMER FOR TICKETS TO ANY EVENT LOGS ONTO THE INTERNET, OR OTHER SUCH WIDE ARE NETWORK, IN HIS CONVENTIONAL MANNER, WHETHER FROM HOME OR OFFICE OR ANY REMOTE LOCATION.
4. SAID PROSPECTIVE CUSTOMER ATTACHES TO THE SERVER THROUGH ANY CONVENTIONAL GRAPHICAL BROWSING MEANS AND VIEWS THE AVAILABLE VENUES, PERFORMANCES, DATES, AND/OR OTHER SUCH OFFERINGS.
5. UPON FIRST ACCESS AN INQUIRY IS MADE TO THE APPROPRIATE DATABASE ASKING FOR A RETURN TO THE PROSPECTIVE CUSTOMER OF ALL GENERALIZED INFORMATION AVAILABLE FROM WHICH HE WILL MAKE A SELECTION OF THE TYPE OF VENUE HE IS SEEKING.
6. SAID PROSPECTIVE CUSTOMER SELECTS A SPECIFIC AREA OF INTEREST FROM THE AVAILABLE VENUES. SUCH MAY BE THEATER TICKETS FOR A PARTICULAR SHOW DATE, AIRLINE SEATS FOR A GIVEN FLIGHT, ETC., AND INDICATES HIS SELECTION THROUGH A CONVENTIONAL HYPERLINK OR OTHER COMPATIBLE MEANS.
7. UPON CONTACT THE SERVER MAKES ANOTHER APPROPRIATE DATABASE QUERY ASKING FOR A RETURN TO THE PROSPECTIVE CUSTOMER OF ALL SPECIFIC INFORMATION RELATING TO HIS SELECTION, I.E., AVAILABLE SEATS FOR THE CHOSEN AIRLINE FLIGHT, FROM WHICH HE WILL MAKE HIS SELECTION OR SELECTIONS.
8. UPON RETURN THE PROSPECTIVE CUSTOMER IS PRESENTED WITH A REPRESENTATION OF ALL AVAILABLE SEATING FOR HIS SELECTED VENUE. FROM THIS REPRESENTATION, WHICH MAY BE GRAPHICAL OR DISPLAYED IN ANY OTHER APPROPRIATE WAY, HE MAKES A SELECTION OF THE SPECIFIC SEAT OR SEATS HE WISHES TO RESERVE AND SUBMITS SUCH TO THE SERVER.
9. THE SERVER CREATES A TEMPORARY CUSTOMER IDENTIFICATION AND ASSOCIATES THE PROSPECTIVE CUSTOMER'S SELECTIONS WITH THAT SPECIFIC IDENTIFICATION SO AS TO PRECLUDE CONFUSION AND ALLOW MULTIPLE SIMULTANEOUS USERS.
10. THE SERVER REQUESTS PAYMENT INFORMATION FROM THE NOW CUSTOMER THROUGH A FORM INPUT MECHANISM. ONCE COMPLETED AND SUBMITTED, THE CUSTOMER'S PAYMENT INFORMATION IS VERIFIED THROUGH CONVENTIONAL MEANS WITH ANY OF THE EXISTING PROCESSING MEANS.
11(A). IF CUSTOMER'S PAYMENT INFORMATION VERIFICATION IS SUCCESSFUL THEN:
(A) THE CUSTOMER'S INFORMATION, AS SUPPLIED IN STEP 10, IS MADE PERMANENT AND;
(B) THOSE SPECIFIC SEATS AS SELECTED BY THE CUSTOMER ARE REMOVED FROM AVAILABLE INVENTORY AND;
(C) MARKED AS UNAVAILABLE ON THE GRAPHICAL REPRESENTATION OF THE VENUE SEATING AS PRESENTED TO THE NEXT PROSPECTIVE CUSTOMER;
(D) ENTERED INTO THE ACCOUNTING AND ADMINISTRATION INFORMATION DATABASE FOR LATER RETRIEVAL BY THE VENUE OPERATOR.
11(B). IF CUSTOMER'S PAYMENT INFORMATION VERIFICATION IS NOT SUCCESSFUL THEN:
(A) CUSTOMER IS GIVEN THE OPPORTUNITY TO CORRECT HIS SUBMITTAL AND TRY AGAIN;
(B) CUSTOMER CAN RETURN TO THE BEGINNING AND REPEAT THE ENTIRE PROCESS.
12. A CONFIRMATION OF THE TRANSACTION, CONTAINING THE TRANSACTION IDENTIFICATION AND OTHER PERTINENT INFORMATION IS RETURNED TO THE CUSTOMER

FIG. 1

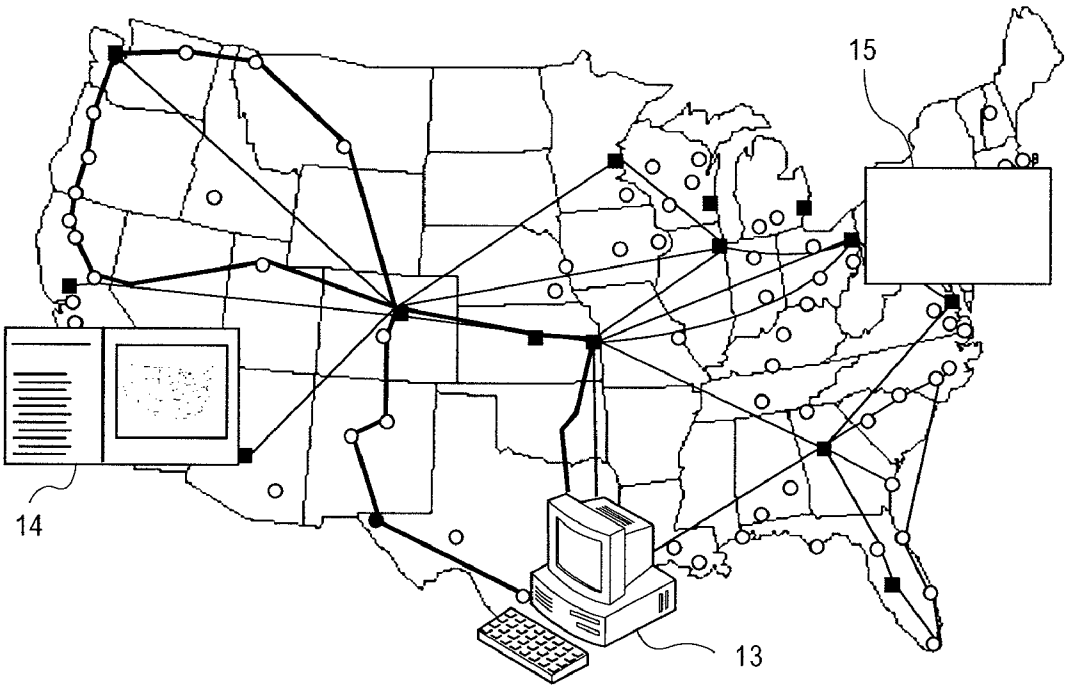


FIG. 2

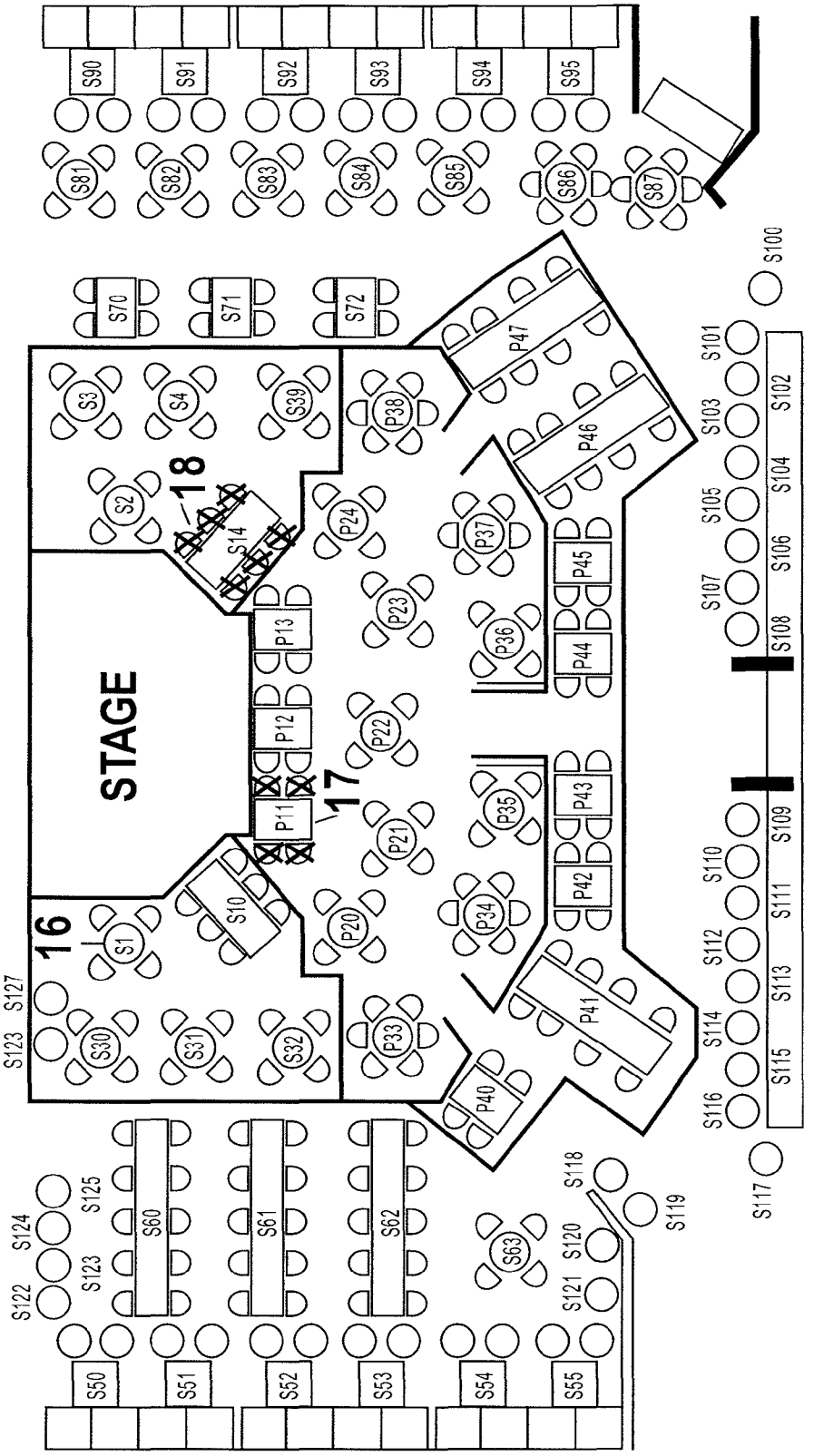

STAGE
16
17
18
S1
S2
S3
S4
S10
S14
S30
S31
S32
S39
S50
S51
S52
S53
S54
S55
S60
S61
S62
S63
S70
S71
S72
S81
S82
S83
S84
S85
S86
S87
S90
S91
S92
S93
S94
S95
S100
S101
S102
S103
S104
S105
S106
S107
S108
S109
S110
S111
S112
S113
S114
S115
S116
S117
S118
S119
S120
S121
S122
S123
S124
S125
S127
P11
P12
P13
P20
P21
P22
P23
P24
P33
P34
P35
P36
P37
P38
P40
P41
P42
P43
P44
P45
P46
P47

FIG. 3A

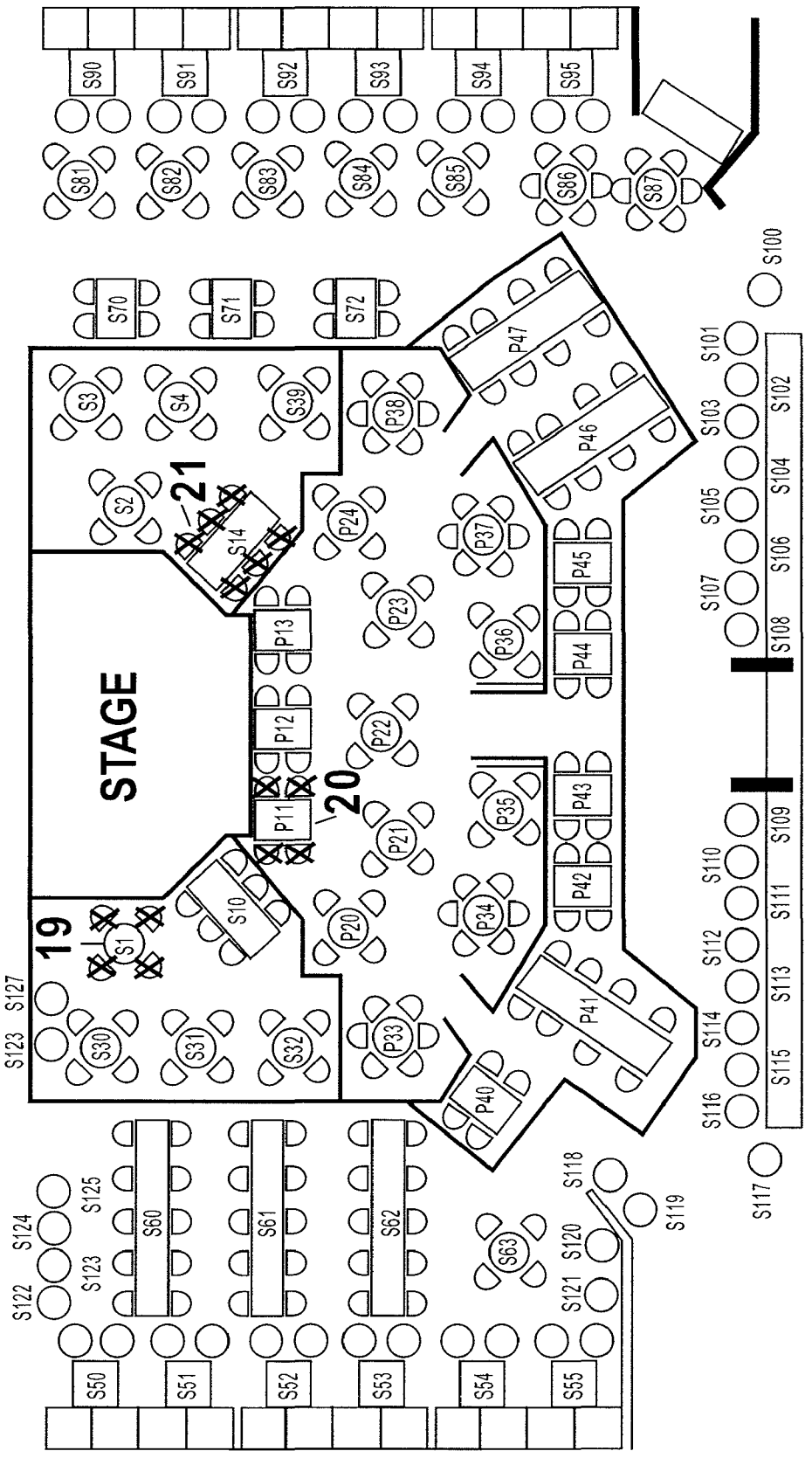

STAGE
19
20
21
S1
S2
S3
S4
S10
S14
S30
S31
S32
S39
S50
S51
S52
S53
S54
S55
S60
S61
S62
S63
S70
S71
S72
S81
S82
S83
S84
S85
S86
S87
S90
S91
S92
S93
S94
S95
S100
S101
S102
S103
S104
S105
S106
S107
S108
S109
S110
S111
S112
S113
S114
S115
S116
S117
S118
S119
S120
S121
S122
S123
S124
S125
S127
P11
P12
P13
P20
P21
P22
P23
P24
P33
P34
P35
P36
P37
P38
P40
P41
P42
P43
P44
P45
P46
P47

FIG. 3B

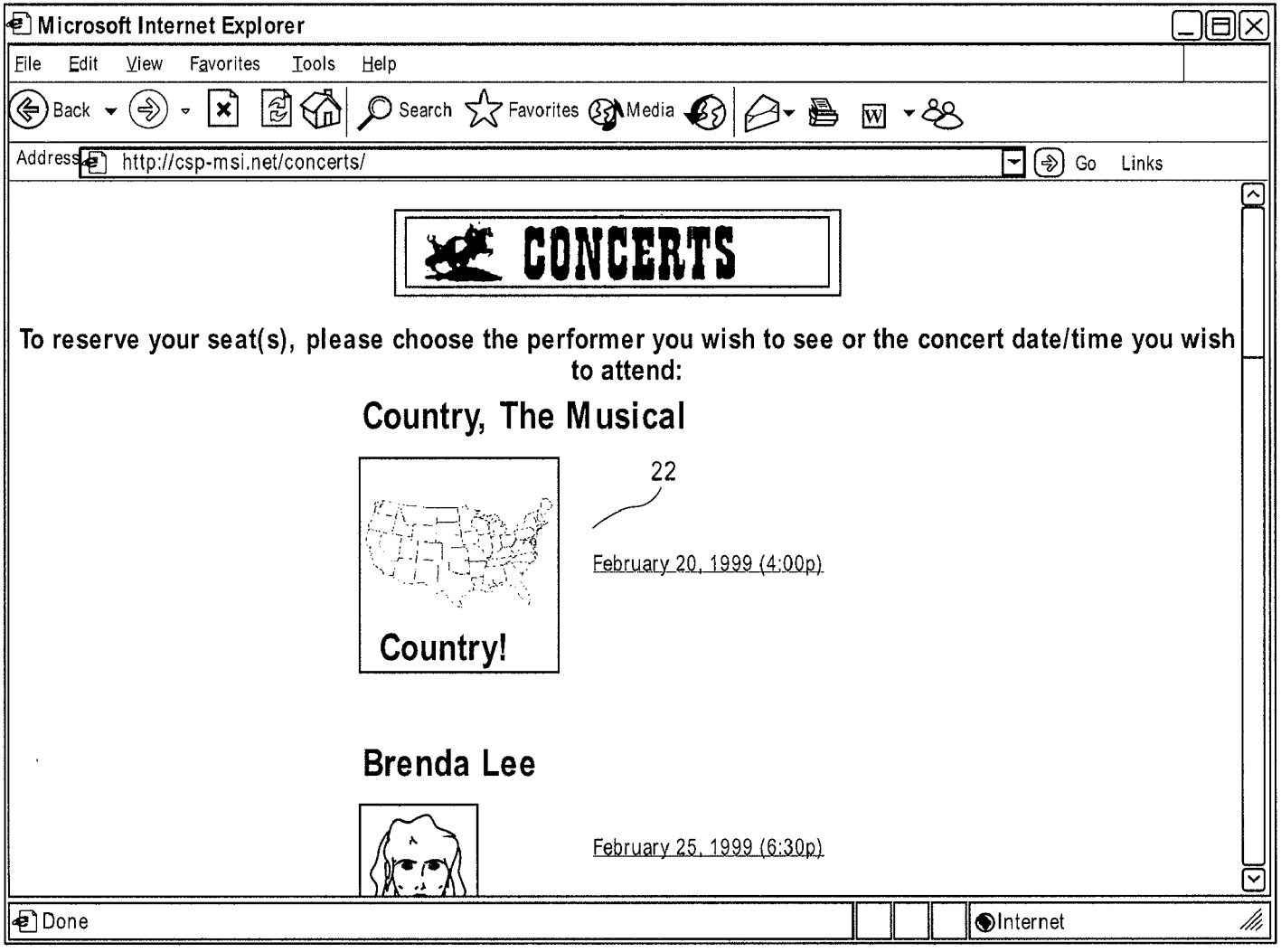

Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://csp-msi.net/concerts/
Go Links
CONCERTS
To reserve your seat(s), please choose the performer you wish to see or the concert date/time you wish to attend:
Country, The Musical
22
February 20, 1999 (4:00p)
Country!
Brenda Lee
February 25, 1999 (6:30p)
Done
Internet

FIG. 4A

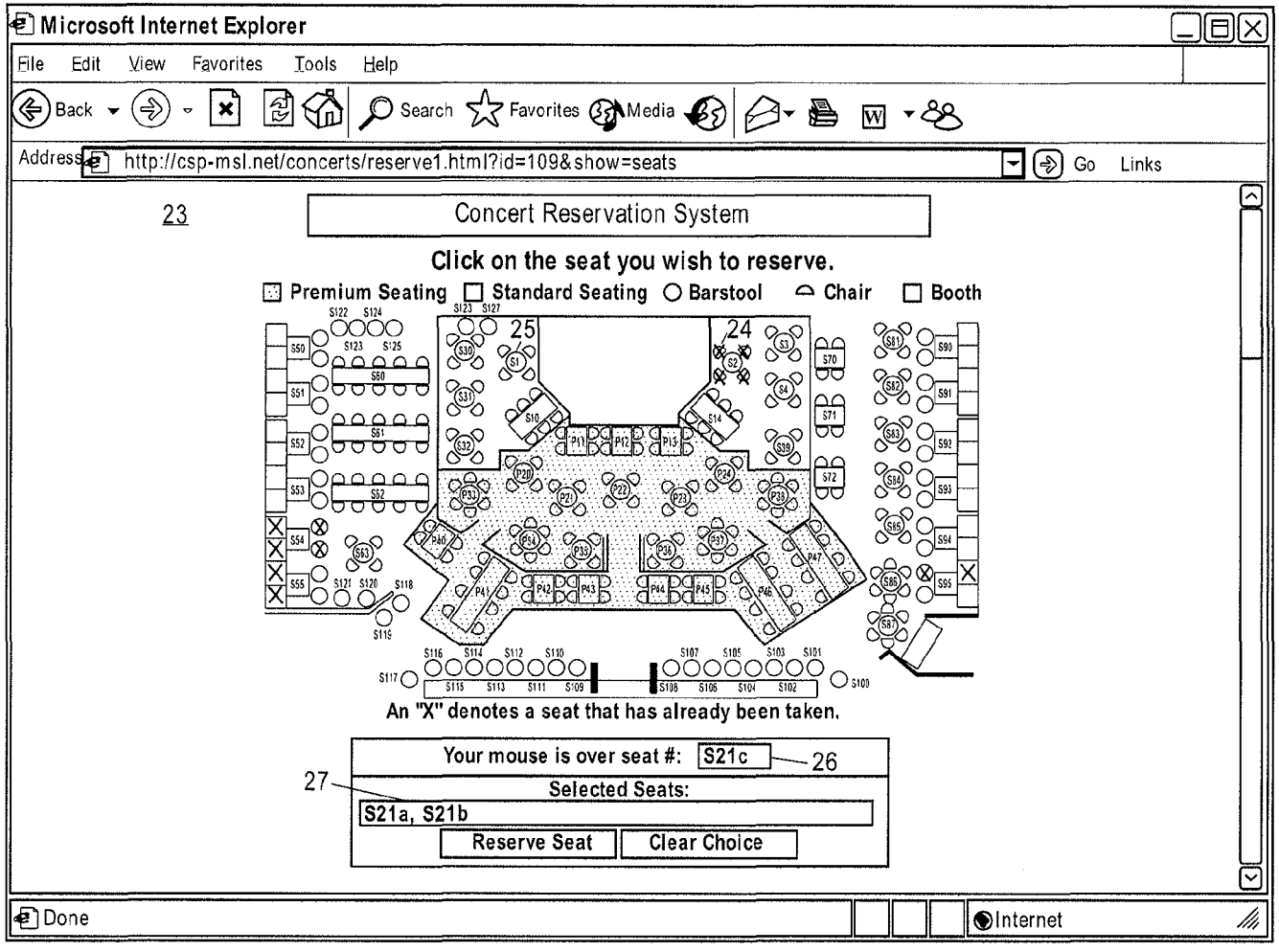

Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://csp-msl.net/concerts/reserve1.html?id=109&show=seats
Go Links
23
Concert Reservation System
Click on the seat you wish to reserve.
Premium Seating
Standard Seating
Barstool
Chair
Booth
25
24
An "X" denotes a seat that has already been taken.
Your mouse is over seat #: S21c
26
27
Selected Seats:
S21a, S21b
Reserve Seat
Clear Choice
Done
Internet

FIG. 4B

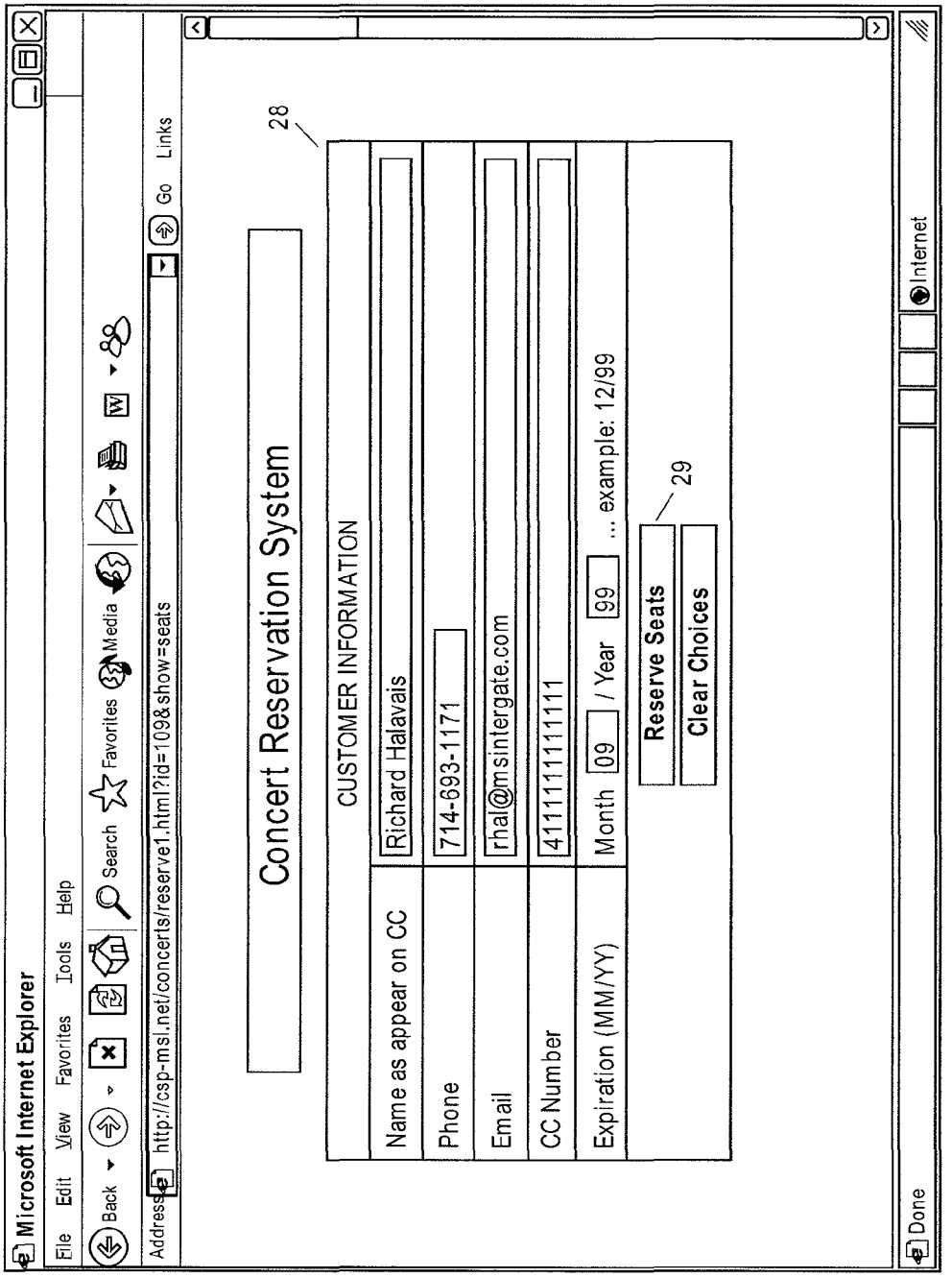

Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://csp-msi.net/concerts/reserve1.html?id=109&show=seats
Go Links
Concert Reservation System
28
CUSTOMER INFORMATION
Name as appear on CC
Richard Halavais
Phone
714-693-1171
Email
rhal@msintergate.com
CC Number
4111111111111
Expiration (MM/YY)
Month 09 / Year 99 ... example: 12/99
Reserve Seats
29
Clear Choices
Done
Internet

FIG. 4C

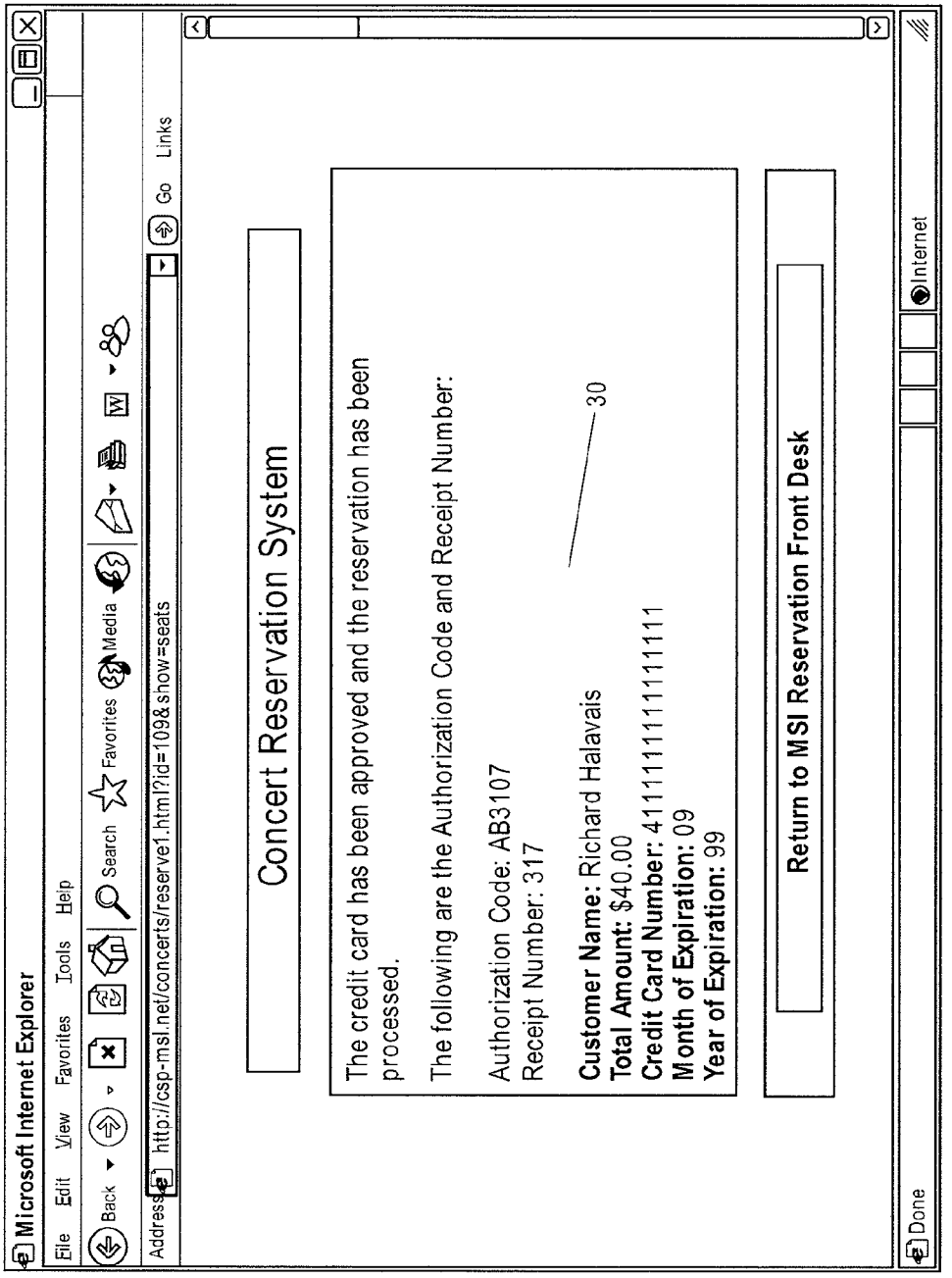

Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://csp-msl.net/concerts/reserve1.html?id=109&show=seats
Go Links
Concert Reservation System
The credit card has been approved and the reservation has been processed.
The following are the Authorization Code and Receipt Number:
Authorization Code: AB3107
Receipt Number: 317
30
Customer Name: Richard Halavais
Total Amount: $40.00
Credit Card Number: 4111111111111111
Month of Expiration: 09
Year of Expiration: 99
Return to MSI Reservation Front Desk
Done
Internet

FIG. 4D

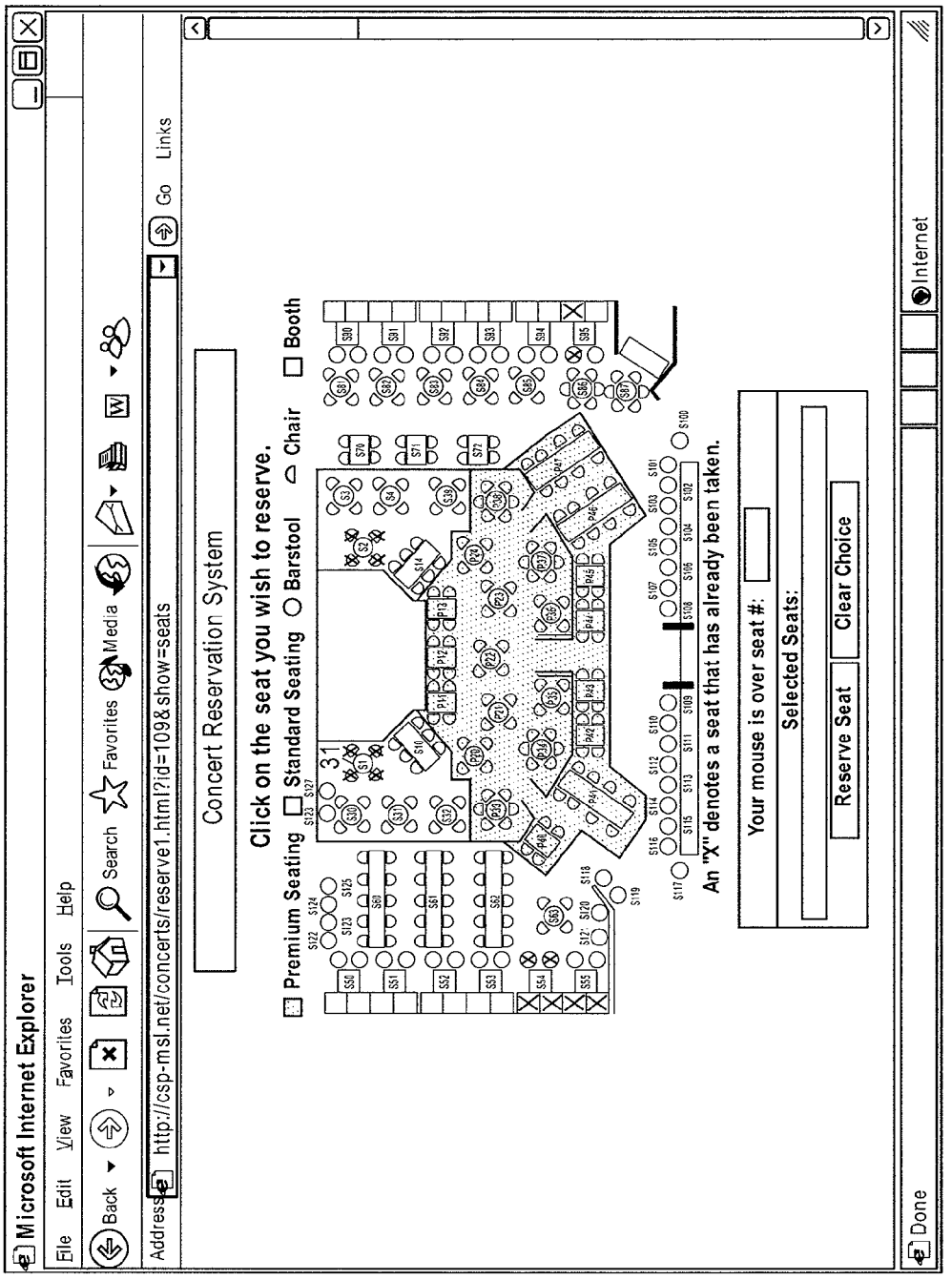

Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media Go Links
Address http://csp-msl.net/concerts/reserve1.html?id=109&show=seats
Concert Reservation System
Click on the seat you wish to reserve.
Premium Seating Standard Seating Barstool Chair Booth
An "X" denotes a seat that has already been taken.
Your mouse is over seat #:
Selected Seats:
Reserve Seat
Clear Choice
Done
Internet

FIG. 4E

FIG. 5

```
Begin Program
    <Markup language>
    <REM --- Imports the file "datasource.inc" which creates variable "datasource" which is
    used to tell markup language datasource to connect to.                    --->
    <INCLUDE NAME="database\datasource.inc">
    <REM--- In case a database or other type of error occurs, this display the error message. -
    -- >
    <ERROR>
    <FONT FACE='Verdana, Arial" SIZE="+1"><B>An Error Has Occurred</B></FONT><P> <FONT
    FACE="Verdana, Arial" SIZE="-1"><B>Error Message = :i_errortext </B></FONT><P>
    <FONT FACE="Verdana, Arial" SIZE="-1"><B>database Error =
    :i_databaseerrortext</B></FONT><P>
    <FONT FACE="Verdana, Arial" SIZE="-1"><B>database Error = i_databaseerrorstmt
    </B></FONT><P>
    </ERROR>
    <REM --- Begin normal markup language here                    --->
    <markup language>
    <HEAD>
    <TITLE>Ticketing &amp;amp; Reservation System</TITLE>
    </HEAD>
    <BODY BACKGROUND="images/background.jpg" TEXT="#000000" LINK="#006666"
    VLINK="#006666">
    <CENTER>
    <IMG SRC="images/masthead_concerts.gif" HEIGHT=60 WIDTH=280><P>
```

```
<FONT FACE="Verdana,Arial". SIZE=-1><B>To begin reserving your seat(s), please select
the concert date/time you wish to attend next to the performer you want to
see:</B></FONT>
<TABLE BORDER=0 CELLSPACING=5 CELLPADDING=5>
<REM --- Begin database query to retrieve all performances that is          --->
<REM --- currently available. Will loop until all available                 --->
<REM --- performing artists and their performances are listed.              --->
<REM--- Part of the return from the query are the links that will           --->
<REM --- take you to the next step of the reservation.                      --->
        <database DBNAME=":datasource"
        database="SELECT id, name, picture, sequence
        FROM category
        WHERE active=1 AND parent=-1
        ORDER BY sequence"
        ALIAS="concert">
        <databaseFETCH ALIAS="concert">
        <WHILE NOTALIAS=i_databaseempty>
        <TR>
        <TO COLSPAN=2>
        <FONT FACE="Verdana,Arial" SIZE=+1><B>:concert_name</B></FONT>
        </TD></TR><TR><TD VALIGN="top">
        <img src=imagesLconcert_picture align=top border="1">
        </TD><TD>
        <Markup language DBNAME=":datasource"
                database="SELECT id, name, date, time
                FROM category
                WHERE active=1 AND parent=rconcert_id
                ORDER BY date, time"
                OUTPUT="<FONT FACE="Verdana,Arial" SIZE=-1><B>
```

FIG. 5
(CONT.)

FIG. 5
(CONT.)

```
                    <A HREF='Reserve1.ihtml?&id=;1'>:2</A>
                    </B></FONT><p">
                </TD></TR><TR><TD COLSPAN=2><HR></TD></TR>
                <databaseFETCH ALIAS="concert">
                </WHILE ALIAS=i_databaseempty>
            </database ALIAS="concert">
            <'TABLE>
            </BODY>
            </markup language>
processing then passes to:
            <Markup language>
            <REM--- Imports the file "datasource.inc" which creates the          --->
            <REM --- variable "datasource" which is used to tell Markup language which          --->
            <REM --- database datasource to connect to.          --->
            <INCLUDE NAME="database\datasource.inc">
            <REM In case a database or other type of error occurs, this will display the error message. >
            <ERROR>
            <FONT FACE="Verdana,Arial" SIZE="+1"><B>An Error Has Occurred</B></FONT><P>
            <FONT FACE="Verdana,Arial" SIZE="-1"><B>Error Message =
            :i_errortext</B></FONT><P>
            <FONT FACE="Verdana,Arial" SIZE="-1"><B>database Error =
            :idatabaseerrortext</B></FONT><P>
            <FONT FACE="Verdana,Arial" SIZE="-1"><B>database Error =
            :i_databaseerrorstmt</B></FONT><P>
            </ERROR>
            <REM --- Begin normal markup language here --- >
            <markup language>
            <HEAD>
            <TITLE>Ticketing &amp;amp; Reservation System - Select Seat(s)</TITLE>
```

FIG. 5
(CONT.)

```
</HEAD>
<REM --- All seats clicked will pass its information to a input box, "newseats". When done,
the information will be passed to "process.ihtml" and be processed by "reserve2.ihtml". --- >
<REM --- This code allows the ability to select multiple seats before proceeding to the next
step of reservation process. Other features include listing the selected seats in the
"number" text box to show the seats that has been clicked. This script also alters the button
text to be grammatically correct. --- >
<SCRIPT LANGUAGE="JavaScript">
<!--
function selectseat(idnum, seatnum)
{
reserved = document. seats. number. value;
if (reserved == '')
{
document. seats. newseats. value = "<Markup language DBNAME=:"+"datasource
database='INSERT INTO basket (custid, pid, qty) VALUES (:"+"custid, "+idnum+",
1)'><database DBNAME=:"+"datasource database='SELECT cost FROM products WHERE
id="+idnum+'"><databaseFETCH><iEQ NAME='ticketprice' VALUE=:"+"1></database><iEQ
NAME='total' VALUE=<iEVAL EXPR=':"+"ticketprice + :"+"total' PREC='2'>><Markup
language DBNAME=:"+" datasource database='INSERT INTO orderdetail (pid, aid, qty, sell)
VALUES ("+idnum+", :"+"oid, 1, :"+"ticketprice)'>";
document.seats.number.value = seatnum;
document. seats. button1. value = "Reserve Seat";
document.seats.button2.value = "Clear Choice";
}
else
{
document. seats.newseats. value += "<Markup language DBNAME=:"+"datasource
database='INSERT INTO basket (custid, pid, qty) VALUES (:"+"custid, "+idnum+",
```

```
1)'><database DBNAME=:"+"datasource database='SELECT cost FROM products WHERE
id="idnum+'"><databaseFETCH><iEQ NAME='ticketprice' VALUE=:"+"1></database><IEQ
NAME='total' VALUE=<iEVAL EXPR=':"+"ticketprice + :"+"total' PREC='2'>><Markup
language DBNAME=:"+"datasource database='INSERT INTO orderdetail (pid, old, qty, sell)
VALUES ("+idnum+", :"+"oid, 1, :"+"ticketprice)'>";
document.seats.number.value = reserved+", "+seatnum;
document. seats. button1. value = "Reserve Seats";
document.seats.button2.value = "Clear Choices";
}
}
function clear()
{
document.seats.newseats.value = '";
}
//-->
</SCRIPT>
<BODY BACKGROUND="images/background.jpg" TEXT="#000000" LINK="#006666"
VLINK="#006666">
<REM --- Begin database query to retrieve the last id number from table "orderdetail" for the
purpose of creating a new instance of the stage image whenever a new order has occurred.
This will prevent the browser from displaying an old floor image from its cache as a different
image name is called every time the file is run. --- >
<database DBNAME=":datasource"
        database="SELECT max(id)
        FROM orderdetail">
        <databaseFETCH>
        <iEQ NAME="imagenum" VALUE=:1>
        <databaseFETCH>
</database>
```

FIG. 5
(CONT.)

```
<iEQ NAME="imagetype" VALUE=".jpg">
<CENTER>
<TABLE BGCOLOR=40000001 BORDER=0 CELLPADDING=5 CELLSPACING=0
WIDTH=500>
<TR><TD><CENTER>
<FONT FACE="Verdana,Arial" SIZE=+1 COLOR='#FFFFFF'><B>
Concert Reservation System
</B></FONT>
</CENTER>
</TD></TR></TABLE></CENTER><p>
<REM - Loads the core image that will be dynamically altered for use as the image map. - >
<iIMAGEFROMFILE NAME="stage" FILENAME="images/stage.jpg" TYPE="jpeg">
<MAP NAME="stage" BORDER=0>
<REM --- Begin database query to retrieve all seat information for the image map
"stage". Will loop until all available seat information for this particular performance is listed.  --->
<REM --- The return query will be used only if the "active" attribute of that particular seat is
marked "true" or "1".  --- >
<REM --- If active, the seat's information, including image map coordinates will be displayed.
Otherwise, the seat on the image map will be x'd out according to the x and y coordinates
associated with it.  --->
<database DBNAME=":datasource"
database="SELECT id, active, x, y, shape, corrds, name, mouseout, mouseover1,
mouseover2, mouseover3
        FROM products
        WHERE catid=:id">
        <databaseFETCH>
        <iWHILE NOTALIAS=i_databaseempty>
        <iEQ NAME="active" VALUE=:2>
        <iCASE ALIAS="active" VALUE=1>
```

FIG. 5 (CONT.)

```
<AREA SHAPE=":5" COORDS=":6" HREF="javascript:selectseat(:1,':7')" ALT="Seat # :7"
OnMouseOut=:8 OnMouseOver=:9 :7 :10 :7 :11 >
        </iCASE ALIAS="active">
        <iCASE ALIAS="active" VALUE=0>
        <iIMAGETEXT NAME="stage" TEXT="X" X=:3 Y=:4 COLOR="red">
        </iCASE ALIAS="active">
        <databaseFETCH>
        </iWHILE ALIAS=i_databaseempty>
</database>
</MAP>
<CENTER>
<FONT FACE="Verdana,Arial" SIZE="-1"><B>
Click on the seat you wish to reserve.
</B></FONT>
<TABLE BORDER=1 CELLPADDING=0 CELLSPACING=0>
<TR><TD>
<REM --- Converts the dynamically generated image, "stage", into a web friendly image type -
"jpeg".          --->
<iIMAGEWRITE NAME="stage" FILENAME="images/ch_stage-:id-:imagenum:"imagetype"
TYPE="jpeg" QUALITY="30">
<REM --- Deletes the temporary image "stage" as it is no longer needed. --- >
<iIMAGEDESTROY NAME="stage">
<IMG SRC="images/legend.jpg" BORDER=0 HEIGHT=20 WIDTH=584><BR>
<IMG SRC="images/ch_stage-:id-imagenum:imagetype" BORDER=0 USEMAP="#stage"
HEIGHT=315 WIDTH=584>
</CENTER>
</TD></TR></TABLE>
<FONT FACE='Verdana,Arial' SIZE=-1><B>
An "X" denotes a seat that has already been taken.
```

FIG. 5
(CONT.)

```
</B></FONT>
<FORM NAME="seats" ACTION="process.ihtml" METHOD="post">
<CENTER>
<TABLE BGCOLOR='#000000' BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR><TD>
<TABLE BGCOLOR='#FFFFFF' BORDER=0 CELLPADDING=5 CELLSPACING=1>
<TR><TD><CENTER>
<FONT FACE='Verdana,Arial' SIZE=-1><B>
Your mouse is over seat#: <INPUT TYPE="text" NAME="seatnum" SIZE=5>
</B></FONT></CENTER>
</TD></TR><TR><TD><CENTER>
<FONT FACE='Verdana,Arial' SIZE=-1><B>
Selected Seats:<BR><INPUT TYPE="text" NAME="number" SIZE=40>
<BR>
<INPUT TYPE="hidden" NAME="newseats" SIZE=50>
<INPUT TYPE="submit" NAME="button1" VALUE="Reserve Seat"> <INPUT
TYPE="reset" NAME="button2" VALUE="Clear Choice" onClick="clear()">
</B></FONT>
</CENTER>
</TD></TR></TABLE></TD></TR></TABLE></CENTER>
</FORM></CENTER></BODY>
</markup language>
```

which then passes to a template:

```
<Markup language>
<REM --- Imports the file "datasource.inc" which creates the variable "datasource" which is
used to tell Markup language which ODBC datasource to connect to.
<#INCLUDE NAME="database\datasource.inc">
<REM --- In case a database or other type of error occurs, this will display the error
message. - >
```

FIG. 5
(CONT.)

```
<ERROR>
<FONT FACE="Verdana,Arial" SIZE="+1"><B>An Error Has Occurred</B></FONT><P>
<FONT FACE="Verdana,Arial" SIZE="-1"><B>Error Message =
:i_errortext</B></FONT><P>
<FONT FACE="Verdana,Arial" SIZE="-1"><B>database Error =
:i_databaseerrortext</B></FONT><P>
<FONT FACE="Verdana,Arial" SIZE="-1 "><B>database Error =
:i_databaseerrorstmt</B></FONT><P>
</ERROR>
<REM--- Copy a preformatted file, "empty.ihtml" into a new file,          --- >
<REM --- "reserve2.ihtml".                                                 --->
<iCOPYFILE SRC="empty.ihtml" DST="reserve2.ihtml>
<REM --- Appends the information passed from "reseve1.ihtml" to "reserve2.ihtml" for
process. Other html information is also passed.                           --->
<iFILE NAME="reserve2.ihtml" DATA=":newseats" OP="append">
<iFILE NAME="reserve2.ihtml" DATA="<INPUT TYPE='hidden' NAME='total'
VALUE=:total>" OP="append">
<iFILE NAME="reserve2.ihtml" DATA="</FORM>" OP="append">
<iFILE NAME="reserve2.ihtml" DATA="</CENTER>" OP="append">
<iFILE NAME="reserve2.ihtml" DATA="<P>" OP="append">
<iFILE NAME="reserve2.ihtml" DATA="</BODY>" OP="append">
<iFILE NAME="reserve2.ihtml" DATA="</markup language>" OP="append">
<REM --- Once the information is appended into "reserve2.ihtml", it will be automatically
push the page "reserve2.ihtml" to the browser.                            --->
<iREDIR URL="reserve2.ihtml">
```

which then combines information and passes to:

```
<Markup language>
<REM --- Imports the file "datasource.inc" which creates the variable "datasource" which is
used to tell Markup language which ODBC datasource to connect to.         --->
```

FIG. 5
(CONT.)

```
<iINCLUDE NAME="database\datasource.inc">
<REM --- In case a database or other type of error occurs, this will display the error
message. - >
<ERROR>
<FONT FACE="Verdana,Arial" SIZE="+1"><B>An Error Has Occurred</B></FONT><P>
<FONT FACE="Verdana,Arial" SIZE="-1"><B>Error Message =
:i_errortext</B></FONT><P>
<FONT FACE="Verdana,Arial" SIZE="-1"><B>database Error =
:idatabaseerrortext</B></FONT><P>
<FONT FACE="Verdana,Arial" SIZE="-1"><B>database Error =
:i_databaseerrorstmt</B></FONT><P>
</ERROR>
<REM --- Copy a preformatted file, "empty.ihtml" into a new file, "reserve2.ihtml".  --->
<iCOPYFILE SRC="empty.ihtml" DST="reserve2.ihtmr>
<REM Appends the information passed from "reseve1.ihtml" to "reserve2.ihtml" for
process. Other html information is also passed.  --- >
<iFILE NAME="reserve2.ihtml" DATA=":newseats" OP="append">
<iFILE NAME="reserve2.ihtml" DATA="<INPUT TYPE='hidden' NAME='total
VALUE=:total>" OP="append">
<iFILE NAME="reserve2.ihtml" DATA="</FORM>" OP="append">
<iFILE NAME="reserve2.ihtml" DATA="</CENTER>" OP="append">
<iFILE NAME="reserve2.ihtml" DATA="<P>" OP="append">
<iFILE NAME="reserve2.ihtml" DATA="</BODY>" OP="append">
<iFILE NAME="reserve2.ihtml" DATA="</markup language>" OP="append">
<REM --- Once the information is appended into "reserve2.ihtml", it will be automatically
push the page "reserve2.ihtml" to the browser.
<iREDIR URL="reserve2.ihtml">
then (takes all information from index & reserve one and conbines it to with the information
in empty) then passes to:
```

FIG. 5
(CONT.)

```
<!Markup language>
<REM --- Imports the file "datasource.inc" which creates the variable "datasource" which is
used to tell Markup language which ODBC datasource to connect to. --- >
<#INCLUDE NAME="database\datasource.inc">
<REM -- In case a database or other type of error occurs, this will display the error
message. -- >
<ERROR>
<FONT FACE="Verdana,Arial" SIZE="+1"><B>An Error Has Occurred</B></FONT><P>
<FONT FACE="Verdana,Arial" SIZE="-1"><B>Error Message =
:i_errortext</B></FONT><P>
<FONT FACE="Verdana,Arial" SIZE="-1"><B>database Error =
:i_databaseerrortext</B></FONT><P>
<FONT FACE="Verdana,Arial" SIZE="-1"><B>database Error =
i_databaseerrorstmt</B></FONT><P>
</ERROR>
<REM --- Begin normal markup language here                    --->
<markup language>
<HEAD>
<TITLE>Ticketing &amp;amp; Reservation System - Customer Information</TITLE>
</HEAD>
<BODY BACKGROUND="images/background.jpg" TEXT="#000000" LINK="#006666"
VLINK="#006666">
<REM-------Get Customer Info------->
<CENTER>
<TABLE BGCOLOR='#000000' BORDER=0 CELLPADDING=5 CELLSPACING=0
WIDTH=500>
<TR><TD><CENTER>
<FONT FACE='Verdana,Arial' SIZE=+1 COLOR='#FFFFFF'><B>
Ticketing &amp;amp; Reservation System
```

FIG. 5
(CONT.)

```
</B></FONT>
</CENTER>
</TD></TR>
</TABLE><p>
<FORM ACTION=reserve3.ihtml METHOD=POST>
<TABLE BGCOLOR='#000000' BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR><TD>
<TABLE BGCOLOR='#FFFFFF' BORDER=0 CELLPADDING=5 CELLSPACING=1>
        <TR>
        <TD COLSPAN=2><CENTER><FONT FACE='Verdana,Arial' SIZE=-
1><B>CUSTOMER INFORMATION</CENTER></TD>
        </TR><TR>
        <TD><FONT FACE='Verdana,Arial' SIZE=-1><B>Name as appear on
CC</B></FONT></TD>
        <TD><FONT FACE='Verdana,Arial' SIZE=-1><B><INPUT TYPE="text"
NAME="name" MAXLENGTH=50 SIZE=40></B></FONT></TD>
        </TR><TR>
        <TD><FONT FACE='Verdana,Arial' SIZE=-1><B>Phone</B></FONT></TD>
        <TD><FONT FACE='Verdana,Arial' SIZE=-1><B><INPUT TYPE="text"
NAME="phone" MAXLENGTH=50 SIZE=12></B></FONT></TD>
        </TR><TR>
        <TD><FONT FACE='Verdana,Arial' SIZE=-1><B>Email</B></FONT></TD>
        <TD><FONT FACE='Verdana,Arial' SIZE=-1><B><INPUT TYPE="text"
NAME="email" MAXLENGTH=50 SIZE=40></B></FONT></TD>
        </TR><TR>
        <TD><FONT FACE='Verdana,Arial' SIZE=-1><B>CC Number</B></FONT></TD>
        <TD><FONT FACE='Verdana,Arial' SIZE=-1><B><INPUT TYPE="text"
NAME="ccnum" MAXLENGTH=50 SIZE=40></B></FONT></TD>
        </TR><TR>
```

FIG. 5
(CONT.)

```
        <TD><FONT FACE='Verdana,Arial' SIZE=-1><B>Expiration
(MM/YY)</B></FONT></TD>
        <TD><FONT FACE='Verdana,Arial' SIZE=-1><B>Month <INPUT TYPE="text"
NAME="ccmexp" MAXLENGTH=50 SIZE=2> / Year <INPUT TYPE="text" NAME="ccyexp"
MAXLENGTH=50 SIZE=2> ... example: 12/99</B></FONT></TD>
        </TR><TR>
        <TD COLSPAN=2>
        <CENTER>
        <FONT FACE='Verdana,Arial' SIZE=-1><B>
        <INPUT TYPE="submit" VALUE="Reserve Seats">
        <BR>
        <INPUT TYPE="reset" VALUE="Clear Choices">
        </B></FONT>
        </CENTER>
        </TD></TR>
</TABLE>
</TD></TR></TABLE>
<iEQ NAME="total" VALUE=0>
<iEQ NAME="date" VALUE='<iDATE>'>
<REM --- Begin database command to insert a new customer profile into the database. This
step is primarily for the purpose of obtaining a new customer id to associate this transaction.
The customer's ip and captured and inserted into a new record in the "customers" table.
The marker, "new" is flagged "true" for the attrieval of the new id. Once the new customer id
is captured, the marker "new" is turned off.                                    --->
<Markup language DBNAME=:datasource
        database="INSERT INTO customers (ip, new)
        VALUES (':i_ip',1)">
<database DBNAME=:datasource
        database="SELECT id
```

FIG. 5
(CONT.)

```
        FROM customers
        WHERE new=1">
        <databaseFETCH>
        <iEQ NAME="custid" VALUE=:1>
</database>
<Markup language DBNAME=:datasource
        database="UPDATE customers
        SET new=0
        WHERE id=:custid">
<REM --- Begin database command to insert a new reservation into the database. This step
is primarily for the purpose of creating a new reservation record, flagged with the newly
obtained customer id, so that the transaction information may be processed in
"resere3.ihtml".                ---->
<Markup language DBNAME=:datasource
        database="INSERT INTO orders (orderdate, approvalcode, receiptnum, totalcharge,
custid)
        VALUES (':date','0000', '0000', :total, :custid)"
FAILURE="The order could not be processed at this time due to technical difficulties.">
<REM --- Begin database command to obtain the reservation id that was just created. This
information, coupled with the customer id, will be used to identify this particular transaction
in the database and be updated in "resere3.ihtml"        --->
        <database DBNAME=:datasource
        database="SELECT max(id)
        FROM orders">
        <databaseFETCH>
        <iEQ NAME="oid" VALUE=:1>
</database>
        <INPUT TYPE="hidden" NAME="reserve" VALUE="receipt">
        <INPUT TYPE="hidden" NAME=custid VALUE=:custid>
```

FIG. 5
(CONT.)

```
<INPUT TYPE="hidden" NAME=phone VALUE=:phone>
<INPUT TYPE="hidden" NAME=email VALUE=:email>
<INPUT TYPE="hidden" NAME=oid VALUE=:oid>
<Markup language DBNAME=:datasource database='INSERT INTO basket (custid, pid, qty)
VALUES (:custid, 6431, 1)'><database DBNAME=:datasource database='SELECT cost
FROM products WHERE id=6431'><databaseFETCH><iEQ NAME='ticketprice'
VALUE=:1></database><iEQ NAME='total' VALUE=<iEVAL EXPR=':ticketprice + :total'
PREC='2'>><Markup language DBNAME=:datasource database='INSERT INTO orderdetail
(pid, oid, qty, sell) VALUES (6431, :oid, 1, :ticketprice)'><Markup language
DBNAME=:datasource database='INSERT INTO basket (custid, pid, qty) VALUES (:custid,
6428, 1)'><database DBNAME=:datasource database='SELECT cost FROM products
WHERE id=6428'><databaseFETCH><iEQ NAME='ticketprice'
VALUE=:1></database><iEQ NAME='total' VALUE=<iEVAL EXPR=':ticketprice + :total'
PREC='2'>><Markup language DBNAME=:datasource database='INSERT INTO orderdetail
(pid, aid, qty, sell) VALUES (6428, :oid, 1, :ticketprice)'><Markup language
DBNAME=:datasource database='INSERT INTO basket (custid, pid, qty) VALUES (:custid,
6429, 1)'><database DBNAME=:datasource database='SELECT cost FROM products
WHERE id=6429'><databaseFETCH><EQ NAME='ticketprice'
VALUE=:1></database><iEQ NAME='total' VALUE=<iEVAL EXPR=':ticketprice + :total'
PREC='2'>><Markup language DBNAME=:datasource database='INSERT INTO orderdetail
(pid, oid, qty, sell) VALUES (6429, :oid, 1, :ticketprice)'><Markup language
DBNAME=:datasource database='INSERT INTO basket (custid, pid, qty) VALUES (:custid,
6430, 1)'><database DBNAME=:datasource database='SELECT cost FROM products
WHERE id=6430'><databaseFETCH><iEQ NAME='ticketprice'
VALUE=:1></database><iEQ NAME='total' VALUE=<iEVAL EXPR=':ticketprice + :total'
PREC='2'>><Markup language DBNAME=:datasource database='INSERT INTO orderdetail
(pid, aid, qty, sell) VALUES (6430, :oid, 1, :ticketprice)'><INPUT TYPE='hidden'
NAME='total' VALUE=:total></FORM></CENTER><P></BODY></markup language>
```

takes in credit card information and passes to:

FIG. 5 (CONT.)

```
<!Markup language>
<REM --- Imports the file "datasource.inc" which creates the variable "datasource" which
is used to tell Markup language which ODBC datasource to connect to.
<iINCLUDE NAME="database\datasource.inc">
<REM -- In case a database or other type of error occurs, this will display the error
message. – >
<ERROR>
<FONT FACE="Verdana,Arial" SIZE="+1"><B>An Error Has Occurred</B></FONT><P>
<FONT FACE="Verdana,Arial" SIZE="-1"><B>Error Message =
:i_errortext</B></FONT><P>
<FONT FACE="Verdana,Arial" SIZE="-1"><B>database Error =
:i_databaseerrortext</B></FONT><P>
<FONT FACE="Verdana,Arial" SIZE="-1"><B>database Error =
:i_databaseerrorstmt</B></FONT><P>
</ERROR>
<REM --- Begin normal markup language here ---->
<markup language>
<HEAD>
<TITLE>Ticketing &amp;amp; Reservation System - Confirmation</TITLE>
</HEAD>
<BODY BACKGROUND="images/background.jpg" TEXT="#000000" LINK="#006666"
VLINK="#006666">
<REM --- Checks to see if the instance "reserve" with a value of ---->
<REM --- "receipt" is passed to this page. If so, the codes within --- >
<REM --- the iCASE and /iCASE tags are executed. This is primarily --- >
<REM --- for security purposes as the instanced "reserved" is only --- >
<REM --- called in the step immediately before. If a visitor ---->
<REM --- accidentally stumbled upon this file, nothing will be run ---->
<REM --- unless all previous steps are completed. ---->
```

FIG. 5
(CONT.)

```
<iCASE ALIAS="reserve" VALUE="receipt">
<CENTER>
<TABLE BGCOLOR='#000000' BORDER=0 CELLPADDING=5 CELLSPACING=0
WIDTH=500>
<TR><TD>
<CENTER>
<FONT FACE='Verdana,Arial' SIZE=+1 COLOR='#FFFFFF'><B>
Concert Reservation System
</B></FONT>
</CENTER>
</TD></TR>
</TABLE>
</CENTER><p>
<REM --- Two steps are combined here:                    --->
<REM --- 1. The reservation information captured in      --->
<REM --- "reserve2.ihtml" is passed to the iPAY tag which    --->
<REM --- processes the information and passes it to CyberCash  --->
<REM --- for verification. If successful, the tag returns  --- >
<REM --- "success", otherwise it returns a variation of "fail"  --- >
<REM --- 2. iIF checks to see if the verification is a success or --- >
<REM --- failure. If successful, it will process the codes  --->
<REM --- immediately after it, which includes updating the  --->
<REM --- database with a successful reservation. If anything  --->
<REM --- other than "success" is returned, iIF will skip to  --->
<REM --- the iELSE tag and execute all codes thereafter.  --->
<iIF COND=<iPAY SERVER="C3"
                amount=:total
                id=:oid
                ccnum=":ccnum"
```

FIG. 5
(CONT.)

```
                    ccmexp=":ccmexp"
                    ccyexp=":ccyexp"
                    name=":name"
                    capture="false"
                    HOST="http://cr.cybercash.com/cgi-bin"
                    PORT=80
                    SECRET="vendorid-26"
                    CRYPTOKEY="j1y1o1ohNU1 ciTdPF1 hsvHFjlpnCpR"»
<CENTER>
<TABLE BGCOLOR='#000000' BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR><TD>
<TABLE BGCOLOR='#FFFFFF' BORDER=0 CELLPADDING=5 CELLSPACING=1
WIDTH=500>
<TR><TD>
<FONT FACE="Verdana,Arial" SIZE="-1">
<B>The credit card has been approved and the reservation has been processed.<P>
The following are the Authorization Code and Receipt Number:</B><p>
<b>Authorization Code:</b> :i_pay_authnumber<BR>
<b>Receipt Number: </b> :ipay_transactionnumber<P>
<b>Customer Name:</b> :name<BR>
<b>Total Amount:</b> $:total<BR>
<b>Credit Card Number:</b> :ccnum<BR>
<b>Month of Expiration:</b> :ccmexp<BR>
<b>Year of Expiration:</b> :ccyexp<BR>
<REM --- Begin database query to update the "customers" table with customer information
captured in "reserve2.ihtml". --->
<Markup language DBNAME=:datasource
                    database="UPDATE customers
                    SET contact=':name', phone=':phone, email=':email'
```

FIG. 5
(CONT.)

```
                WHERE id=:custid">
<REM --- Begin database query to update the "orders" table is updated with the approval
code returned by CyberCash, as well as the transaction information (total charge, credit
card number, etc).                                    --->
<Markup language DBNAME=:datasource
                database="UPDATE orders
                SET approvalcode=':i_pay_authnumber',
receiptnum=':i_pay_transactionnumber', totalcharge=:total,cc=':ccnum', ccm=':ccmexp',
ccy=':ccyexp'
                WHERE id=:oid"
        FAILURE="The information was not saved correctly.<br>">
<REM --- Begin database query to turn off the availability of the seats that have been
reserved by setting the "active" attribute of the seats to "0".                    --->
<database ALIAS="markoff" dbname=":datasource" database="SELECT pid FROM
orderdetail WHERE oid=:oid">
        <databaseFETCH ALIAS="markoff">
        <iWHILE NOTALIAS=i_databaseempty>
                <Markup language dbname=":datasource"
                        database="UPDATE products
                        SET active=0
                                WHERE id=:markoff_pid">
                        <databaseFETCH ALIAS="markoff">
                </iWHILE ALIAS=i_databaseempty>
</database ALIAS="markoff">
</FONT>
</TD></TR>
</TABLE>
</TD></TR>
</TABLE>
```

FIG. 5
(CONT.)

```
</CENTER><p>
<iELSE>
<CENTER>
<TABLE BGCOLOR='#000000' BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR><TD>
<TABLE BGCOLOR='#FFFFFF' BORDER=0 CELLPADDING=5 CELLSPACING=1
WIDTH=500>
<TR><TD>
<FONT FACE="Verdana,Arial" SIZE="-1"><B>
This transaction could not be processed by Cyber Cash at this time. Either the Credit Card
Information was invalid or the Cyber Cash server is currently not available.
</B></FONT>
</TD></TR>
</TABLE>
</TD></TR>
</TABLE>
</CENTER>
</iIF>
<FORM ACTION="index.ihtml" METHOD="post">
<CENTER>
<TABLE BGCOLOR='#000000' BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR><TD>
<TABLE BGCOLOR='#FFFFFF' BORDER=0 CELLPADDING=5 CELLSPACING=1
WIDTH=500>
<TR><TD>
<CENTER>
<FONT FACE="Verdana,Arial" SIZE="-1"><B>
<INPUT TYPE="submit" VALUE="Return to Concert Selection">
</B></FONT>
```

FIG. 5
(CONT.)

```
</CENTER>
</TD></TR>
</TABLE>
</TD></TR>
</TABLE>
</CENTER>
</FORM>
<FORM ACTION="http://domainname/filename" METHOD="post">
        <CENTER>
<TABLE BGCOLOR= 000000' BORDER=0 CELLPADDING=0 CELLSPACING=0> <TR><TD>
<TABLE BGCOLOR='#FFFFFF' BORDER=0 CELLPADDING=5 CELLSPACING=1
WIDTH=500>
<TR><TD>
<CENTER>
<FONT FACE="Verdana,Arial" SIZE="-1"><B>
<INPUT TYPE="submit" VALUE="Return to On-Line Ticketing & Reservation Front Desk">
</B></FONT></CENTER>
</TD></TR><TTABLE></TD></TR></TABLE></CENTER>
</FORM>
</CASE ALIAS="reserve">
</BODY>
</markup language>
```

which verifies and confirms payment information

FIG. 5
(CONT.)

SYSTEM AND METHOD FOR MAINTAINING COHERENCY OF DATA ENTRIES

CROSS REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 12/244,727 filed on Oct. 2, 2008, which is a continuation U.S. patent application Ser. No. 09/295,577 filed on Apr. 22, 1999, now U.S. Pat. No. 7,454,361, the disclosures of which are hereby incorporated by reference in their entirety.

BACKGROUND

1. Field of the Invention

The present invention relates to system and method for maintaining coherency of data entries related to seat availability information obtained from remote venue locations.

2. Prior Art

The inadequacies and inefficiencies of present ticketing and reservation systems are recognized and addressed through this invention. The rapid growth of the internet now makes it possible for anyone at his home or office to avail himself of the advantages of the instant invention through a simple internet or other wide area network connection.

Prior art makes a feeble attempt to address the inconvenience of ticket ordering. U.S. Pat. No. 5,797,126 (1988), Helbling, et al., describes a series of individual kiosks in wireless communication with a central station where a visitor can locate events of interest, view an excerpt of scenes from that venue and purchase tickets. This falls far short of the instant invention since it still requires a user to physically visit a remote site to avail himself of the service. Additionally, said prior art makes extensive use of what is called "kiosks" implying that, unlike the instant invention, it is only from his specialized machines that such services may be rendered.

U.S. Pat. No. 4,974,252 describes a more interactive theater attendance system where patrons are permitted two way communications between themselves and a broadcast center but this is still far from objectives of the instant invention and requires that persons be in attendance at the theater and, further some attendant be present at the remote broadcast center. The instant invention is fully automated and, other than the normal monitoring of any application for a wide area network, requires no human attendance or intervention.

U.S. Pat. No. 3,427,438 describes a ticket vending system where sales of tickets can be recorded on a seating layout but, again falls far short of the instantaneous update and automatic operation of the instant invention. U.S. Pat. No. 5,333,257 allows for a view from a seat but that is now common for internet applications where a hyperlink to any graphic is routinely provided and ancillary to and even unnecessary to the instant invention. Other prior art does nothing to make ticket ordering or seating reservations more readily available and does nothing to improve the information flow to prospective customers so that they may make a more informed decision about attending any given event.

Consider the traveler who has planned a vacation in, say, New York City and wants tickets for some Broadway show. Presently he has either to phone and accept someone else's definition of what constitutes "best available" or wait until he gets into town and seek out a scalper or reseller agency and he still isn't sure exactly what his seats offer.

OBJECTS AND ADVANTAGES

Several objects and advantages of the present invention are:

(a) to always provide customers with a seat selection comprised of the total of the then best available seats for any given event;

(b) to make equally available to all customers all then available seats for any given event to that said customer, so he may select for himself the seat or seats he wants for that event;

(c) to provide an alternative means to visiting box office or a ticket reseller for a customer to select and reserve for himself the then best available seat or seats for any arena, stadium, theater, airline flight or any other such venue where seating is available;

(d) to provide to a ticket sensitive venue operator better control over the seating and seat availability for the various events he offers;

(e) to provide to a ticket sensitive venue operator better accounting of his seating for the various events he offers;

(f) to provide to the general public a more cost effective means by which he may reserve and buy tickets for any given event;

(g) to provide to a venue operator a more cost effective means by which he may sell tickets for any given event;

(h) to provide to the general public an automated 24 hours a day, seven days a week means by which he may reserve and purchase the specific seats he wants for an event;

(i) to provide to a venue operator an automated 24 hours a day, seven days a week means by which he may offer reservation and purchase of a seat or seats that are individually selectable by a customer;

(j) to permit the venue operator to avoid overbooking an event;

(k) to permit the venue operator to avoid underbooking an event;

(l) to permit a doctor or other professional for whom his time is a commodity to better schedule and regulate his time and interact with patients or others in the online environment;

(m) to provide to the general public a 24 hours a day, seven days a week means by which they may schedule appointments with doctors, dentists, automotive mechanics and the like with full knowledge that the appointment time they select is still available.

Further objects and advantages are to provide any venue operator the benefit of better control over his ticket inventory and sales such as to improve his profitability and the efficiency of his operation and to provide to the general ticket buying public better information and an easier means by which they may acquire their desired ticket or tickets for any event.

SUMMARY OF THE INVENTION

The ticketing and reservation system of the present invention, in one particular embodiment thereof, includes a computer program operating on a server for a wide area network (WAN), generally described by the flow chart of FIG. **1** and the accompanying code example which implements the instant invention in practice:

First, when a user accesses the system means is provided to initialize the process and return to the user a menu from which he selects his venue of interest. This can be a selectable menu arranged by artist or date or time or specific theater or football team or baseball team or name or activity or any combination thereof such that the user is given sufficient information from which to make a decision. An example would be someone looking for the next event at a given theater at a time that starts at 7:00 pm. One of many possible series of computer instructions to perform this function is:

<—Send database query to retrieve all venues that are currently available in the system—>
<—Server receives and processes query—>
<—Query is looped until all available performances and venues are retrieved—>
<—Markup Language engine converts result to display compatible format for output to client computer—>
<—Begin normal markup language here—>
<—Begin reservation process selecting the event date/time next to the desired venue—>

THEN, upon user submittal, the server initializes the process of returning to the user all available seats:

<—Send database query to retrieve all seats that are currently available in the system for this event—>
<—Server receives and processes query—>
<—Query is looped until all available seats are retrieved—>
<—Markup language engine converts result to markup language format for output to client computer—>
<—Begin normal markup language here—>
<—Continue reservation process by selecting the desired seat or seats—>

THEN, upon user submittal we create a new customer record in the Wide Area Network server and tell the system which database to connect to fulfill the user's request(s):

<—Send database command to insert new record in customer database and obtain record id—>
<—Send database command to insert new record in reservation "order" database and obtain record id—>
<—Send database command to insert new record for each selected seat in the reservation "detail" database—>
<—Begin normal markup language here—>
<—Continue reservation process by requesting client payment information—>

THEN, upon user submittal the information is passed for verification:

<—Submit client information for verification—>
<—If verification is successful, send database command to update customer record in customer database with information previously collected—>
<—If verification is successful, send database command to update reservation record in reservation "order" database with verification information—>
<—If verification is successful, send database command to remove selected seats from seat inventory database and mark as no longer available for future selection—>
<—Markup language engine converts result to markup language format for output to client computer—>
<—Begin normal markup language here—>
<—If verification is successful, confirmation is generated via Markup language engine to markup language format for output to client computer—>
<—If verification is unsuccessful, a failure notice is generated via Markup language engine to markup language format for output to client computer—>
<—If verification is unsuccessful, client is presented with option to provide his payment information again or abandon his reservation—>

While this is one preferred form of the code there are many other code sequences that will perform the same function that will be immediately obvious to one skilled in the art.

BRIEF DESCRIPTION OF THE DRAWINGS

The aforementioned objects and advantages of the present invention as well as additional objects and advantages thereof will be more fully understood hereinafter as a result of a detailed description of a preferred embodiment when taken in conjunction with the following drawings in which:

FIG. **1** is a block diagram of the present invention illustrating the major components thereof and the interactivity that takes place between the potential customer and the instant invention.

FIG. **2** is an illustration of the concept of the present invention utilizing the internet as the Wide Area Network to which users connect to perform the desired function and shows an example of a remotely located user accessing the functionality of the instant invention for purposes of reserving seats for a dinner theater performance in a distant city.

FIGS. **3**A and **3**B are illustrations of the concept of the present invention refined down to the functionality of reserving specific seats and blocking duplicate sale of those seats that are already reserved.

FIGS. **4**A to **4**E are sample screens seen by a remote user of the instant invention during a session wherein he selects and orders four specific seats for a distant dinner theater show.

FIG. **5** is a complete code set for one preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring to FIG. **1**, it will be seen that the operator of a venue implements the instant invention for purposes of allowing remotely located users to reserve specific seating for specific events **1**. By doing so, he initiates those certain actions necessary to display an internet web site to all online users **2**. A prospective customer for the venues offering(s) logs onto the internet **3** and acquires the aforesaid internet web site **4** which implements the instant invention. He can be connected to the internet by any conventional means yet this by no means implies that the wide area network must be what is commonly referred to as the "internet." Upon first contact by the prospective customer, an inquiry is directed to the appropriate database, which may be located concurrent with the primary server hosting the program for the instant invention or may be located remotely, such as at the physical location of the venue, asking for a return of information to the prospective customer of all appropriate information contained therein relative to his inquiry **5**. The prospective customer indicates his desired seat or seats through conventional computer input means and directs that information back to the server hosting the code necessary to the implementation of the instant invention **6**. Upon contact **7** the server again makes an appropriate database query and returns to the prospective customer all pertinent information relating to his selection, such as which seats are still available for the chosen performance, airline flight, boxing match, etc. The prospective customer is then presented with a representation of all then available seating for his selected venue **8**. From this representation, the prospective customer makes his selection of a

seat or seats by indicating such through a mouse click, keyboard entry or other means, such as but not limited to a touch screen. Simultaneously, the server, through the coding necessary to implement the instant invention, creates a temporary customer identification **9** that is used to associate this potential customer with this later selection and permit system use by multiple simultaneous users. Once the customer has made his seat selection he is asked for payment information **10**. That information is processed through conventional internet or other electronic means and once the information and payment are verified **11***a* the customer information, as supplied in **10**, is made permanent and the seat or seats he has selected are removed from inventory and blocked from duplicate sale, both graphically when presented to the next prospective customer and in the database where information for accounting and administrative purposes is retained. If the customer's payment information cannot be verified **11***b* then he is given an opportunity to correct the information or start over with a new transaction. Upon verification of the customer's payment information he receives a confirmation of the transaction **13** containing all appropriate reference information for his records.

Referring to FIG. **2**, it will be seen that, for example, a user in Houston **13** is planning to vacation in New York and wishes to see a play at a dinner theater there that utilizes the present invention for ticketing and reservations **15**. The user in Houston, or in any other location worldwide, connects to the internet in the conventional way and retrieves the appropriate web site through his graphical browser from a server located in, say, Anaheim, Calif. **14**. Through implementation of the instant invention the user is able to see the exact seating arrangement of the remote dinner theater and select the exact seat or seats he wants for the performance of his choice. Such additional information as is appropriate can be provided to the remote user to assist him in making an informed decision as to which seat or seats he wishes to occupy for this performance.

Referring to FIGS. **3**A and **3**B, it will be seen that in FIG. **3**A that, at the user selected venue, all seats at table p**11** **17** and at table s**14** **18** have been previously taken and are so indicated by the graphical representation of an "X" over those seats. Our potential customer wants to seat a party of four at table s**1** **16** and so indicates by clicking his mouse on those four seats or by so indicating through an alternative standard computer input means. Once his payment method is verified his selected seats are removed from inventory and so indicated on the graphical representation by placing an "X" over those seats **19** while retaining the "X" over those seats previously sold at table p**11** **20** and table s**14** **21**. The next prospective customer is advised that these seats are no longer available for this performance by the new graphical representation shown in FIG. **3**B that is his first viewing screen upon entry into the system. In the event that two prospective customers wish to reserve the exact same seat or seats, that prospective customer who first receives validation of his payment method is given those seats while the other prospective customer is notified that while making his decision the seats he wants have already been sold and offers him a chance to select other seating.

Referring to FIGS. **4**A to **4**E, one will see the screens presented to a user when he accesses the system and as he progresses through the process of selecting a specific seat or seats, then reserves and pays for them. FIG. **4**A is where the first screen presented shows links to available performances for that selected venue **22**. FIG. **4**B is the second screen **23** and shows a view of the seating available for that venue with seats that have already been taken crossed off with an "X" **24**. Our hypothetical user decides that he would like to have his party of four sit at table s**11** **25** and selects the four seats around that table by clicking on them with his mouse. As he moves his mouse over individual seats the seat number appears in the window at the bottom of his screen **26** and when he clicks on a seat it is added to a running tally of the seats he has already taken **27**. Only seats that have not previously been taken show up in the mouse-over window **26**. After completing his selections the user clicks on the "Reserve Seats" button and FIG. **4**C shows his next screen, which asks him for his payment information **28**. He enters the required information and again clicks the "Reserve Seats" button **29**. FIG. **4**D is the next screen and it tells him that his payment method has been accepted (or rejected) and relates information about the transaction **30** such as his transaction code and the receipt number that he can use as a ticket or as a voucher with which to redeem his ticket or tickets at the venue box office when he arrives for the performance. Finally, FIG. **4**E shows the opening screen the next visitor to the system is presented with the same set of screens except that the seats reserved by our hypothetical user **31** are marked off as already taken.

Referring to FIG. **5**, one will see one of many possible coding schemes making possible the results of the present invention.

Those having skill in the art to which the present invention pertains will now understand that there are virtually unlimited uses for the present invention. By way of example, the present invention may be readily used to reserve specific seats on commercial airliners or reserve specific staterooms on a cruise ship, as well as for reserving seats for any venue from community theater or little league baseball to major league sporting events.

The present invention has been described in sufficient detail to enable one skilled in the art to make and use the invention. Accordingly, specific details which are readily available in the art or otherwise conventional have been omitted to prevent obfuscation of the essential features of the invention.

In view of the foregoing it will be understood that the present invention may be implemented in a variety of alternative methods but that all such implementations are deemed to be within the scope of the present invention which is to be limited only by the claims appended hereto:

What is claimed is:

1. A computer-implemented method for maintaining coherency of data entries related to a database that relates to an inventory for a venue and providing access to a plurality of users to the database, the method comprising:

receiving, from a first general purpose computer, a query for information from the database regarding a status of one or more seats in the venue;

determining information indicating an availability of the one or more seats in the venue based on parameters specified in the query;

transmitting, to the first general purpose computer, a response to the query that includes information to be displayed to a first user, wherein the information to be displayed to the user comprises an interactive set of user interface elements that allow the first user to select one or more seats in the venue, wherein, in response to the first user placing a mouse over at least a portion of the interactive set of user interface elements, current available seat information in the venue indicating seats that are currently available is transmitted to the first general purpose computer;

receiving, from the first general purpose computer, information indicating selections with a mouse by the first user;

creating a temporary record in the database, for the first user, based on the information indicating selections by the first user;

transmitting, to the first general purpose computer, a request for confirmation of the selections by the first user based on information in the temporary record;

receiving, from the first general purpose computer, the confirmation of the selections by the first user;

locking entries in the database based on the confirmation, wherein the lock prevents a second user from selecting any seat confirmed for the first user; and

transmitting, to a second general purpose computer, information to be displayed to the second user that indicates seats selected by the first user.

2. The method of claim **1** further comprising:

transmitting a portion of the information to the second general purpose computer to be displayed, the portion including updated information indicating the purchased seat as unavailable.

3. The method of claim **1**, wherein the database comprises entries denoting availability of a plurality of venues.

4. The method of claim **1**, wherein the venue is a theater.

5. The method of claim **1**, wherein the venue is a stadium.

6. The method of claim **1**, wherein the venue is an airplane.

7. A computer system, comprising:

a database that relates to an inventory for a venue; and

a server system that provides to a plurality of users access to the database, the server system configured to:

receive, from a first general purpose computer, a query for information from the database regarding a status of one or more seats in the venue;

determine information indicating an availability of the one or more seats in the venue based on parameters specified in the query;

transmit, to the first general purpose computer, a response to the query that includes information to be displayed to a first user, wherein the information to be displayed to the user comprises an interactive set of user interface elements that allow the first user to select one or more seats in the venue, wherein, in response to the first user placing a mouse over at east a portion of the interactive set of user interface elements, current available seat information in the venue indicatinci seats that are currently available is transmitted to the first general purpose computer;

receive, from the first general purpose computer, information indicating selections with a mouse by the first user;

create a temporary record in the database, for the first user, based on the information indicating selections by the first user;

transmit, to the first general purpose computer, a request for confirmation of the selections by the first user based on information in the temporary record;

receive, from the first general purpose computer, the confirmation of the selections by the first user;

lock entries in the database based on the confirmation, wherein the lock prevents a second user from selecting any seat confirmed for the first user; and

transmit, to a second general purpose computer, information to be displayed to the second user that indicates seats selected by the first user.

8. The computer system of claim **7**, wherein the server is further configured to:

transmit a portion of the information to the second general purpose computer to be displayed, the portion including updated information indicating the purchased seat as unavailable.

9. The computer system of claim **7**, wherein the database comprises entries denoting availability of a plurality of venues.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 7,548,867 B2
APPLICATION NO. : 12/329512
DATED : June 16, 2009
INVENTOR(S) : Halavais et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In Figure 1, sheet 1 of 30, at line 6, change “ARE” to --AREA--.

In Figure 5, sheet 12 of 30, at line 7, change “<‘Table>” to --</Table>--.

In Figure 5, sheet 13 of 30, at line 2, change “a input” to --an input--.

In Figure 5, sheet 13 of 30, at line 16, change ““+idnurn+”,” to --“+idnum+”,--.

In Figure 5, sheet 14 of 30, at line 4, change “old,” to --oid,--.

In Figure 5, sheet 15 of 30, at line 22, change “corrds,” to --coords,--.

In Figure 5, sheet 18 of 30, at line 13, change ““reseve1.ihtml”” to --“reserve1.ihtml”--.

In Figure 5, sheet 19 of 30, at line 14, change “.ihtmr>” to --.ihtml”>--.

In Figure 5, sheet 19 of 30, at line 15, change ““reseve1.ihtml”” to --“reserve1.ihtml”--.

In Figure 5, sheet 19 of 30, at line 28, change “conbines” to --combines--.

In Figure 5, sheet 22 of 30, at line 22, change “ip and” to --ip is--.

In Figure 5, sheet 22 of 30, at line 23, change “attrieval” to --retrieval--.

In Figure 5, sheet 23 of 30, at line 13, change ““resere3.ihtml”.” to --“reserve3.ihtml”.--.

In Figure 5, sheet 23 of 30, at line 21, change ““resere3.ihtml”” to --“reserve3.ihtml”--.

In Figure 5, sheet 27 of 30, at line 19, change “:ipay_” to --:i_pay_--.

In column 3, at line 36, change “connect to fulfill” to --connect to to fulfill--.

In column 5, at line 67, change “s11” to --s1--.

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 7,548,867 B2
APPLICATION NO. : 12/329512
DATED : June 16, 2009
INVENTOR(S) : Halavais et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In column 8, at line 5, in Claim 7, change "at east" to --at least--.

In column 8, at line 7, in Claim 7, change "indicatinci" to --indicating--.

Signed and Sealed this

Fifteenth Day of September, 2009

David J. Kappos

David J. Kappos
Director of the United States Patent and Trademark Office