PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ^{6} :
G06F 17/30, 17/24, 153/00

A2

(11) International Publication Number: WO 99/63460

(43) International Publication Date: 9 December 1999 (09.12.99)

(21) International Application Number: PCT/US99/09319

(22) International Filing Date: 29 April 1999 (29.04.99)

(30) Priority Data:
60/083,593 30 April 1998 (30.04.98) US

(71) Applicant *(for all designated States except US)*: EC CUBED, INC. [US/US]; Suite 310, 15 River Road, Wilton, CT 06897 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: HOQUE, Faisal [US/US]; 96 Glenbrook Road #38, Stamford, CT 06611 (US). KRISHNAMURTHY, Tharak [IN/US]; 7222 Avalon Gate, Trumbull, CT 06611 (US). SHARMA, Tarun [IN/US]; 31 Soundview Avenue #1, Stamford, CT 06902 (US).

(74) Agent: COHEN, Neil, G.; Cummings & Lockwood, Four Stamford Plaza, Stamford, CT 06904 (US).

(81) Designated States: CN, IN, JP, RU, US, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: SYSTEM AND METHOD FOR MANAGING DOCUMENTS AVAILABLE TO A PLURALITY OF USERS

(57) Abstract

A system and method for managing access to a document by a plurality of end users is disclosed. The system and method includes storing the document in a memory of a server. According to a disclosed method, steps include transmitting a first copy of the document to a computer of a first user and transmitting a second copy of the document to a computer of a second user. In addition, receiving a modified first copy of the document from the computer of the first user and storing the modified first copy of the document in the memory under a first document name. And further, receiving a modified second copy of the document from the computer of the second user and storing the modified second copy of the document in the memory under a second document name. And still further, storing data identifying the modified first copy of the document and data identifying the modified second copy of the document in the memory.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

SYSTEM AND METHOD FOR MANAGING DOCUMENTS AVAILABLE TO A PLURALITY OF USERS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to providing a plurality of end users with access to a document on a computer network, and more particularly, to a system and method enabling a plurality of end users to access, modify, and store a document on a computer server of a computer network.

2. Background of the Invention

The conduct of business has been greatly impacted by the rapid advances in computer technology and related communications technology. These technologies have made it substantially easier and less time consuming to conduct daily interactions between businesses. For example, facsimile machines have enabled a buyer to quickly transmit a request for quotations to its sellers and have enabled the sellers to just as quickly return a response. Formerly, when such bid cycles were transacted by mail, they would take days or even weeks to complete. Currently, bid cycles of this type can be accomplished within hours. As a result of the advances in technology, costs in transacting business have been reduced, products get to market more rapidly, and resources have been freed up for other opportunities.

Even with these improvements, however, it is not unusual for the information that is transferred electronically (e.g., by facsimile) to be repeatedly printed onto paper. With respect to the bidding cycle described above, a buyer will typically prepare a request for a quotation on a word processor and print it onto paper prior to transmitting it to several sellers. Upon receipt, each seller prints the request for a quotation onto paper so that they can prepare a quotation. After the sellers prepare their respective quotations, they print them onto paper prior to transmitting them to the buyer. And finally, after the quotations are received by the buyer on a facsimile machine, the quotations are again printed onto paper so that they can be evaluated.

The foregoing process, while being much more efficient than the previous all-paper method, is still relatively slow and inefficient. For example, the present process still involves the use of large quantities of paper that is ultimately discarded. The discard of such large amounts of paper is, of course, a waste of natural resources. In addition, it takes a substantial amount of time to handle the printed documents, for example, to obtain printed versions of the documents and, thereafter, to operate the facsimile machine to transmit them. Furthermore, even though the lag time between the previous all-paper method has been greatly improved, there are still substantial lag times built into the foregoing electronic transfer process. For example, documents are often delayed in reaching the intended recipient because they sit in or near the facsimile machine after being "delivered", they are in transit between the facsimile machine and the recipient's desk, or they sit in the recipient's in-box.

Because of the time required for the completion of the above described process, considerable lead times are still required in order to ensure that the necessary goods and services are available when required so that a buyer can produce a specific product. The systems and methods of the prior art leave considerable room for improvement in the productivity of both the buyer and the seller by requiring such relatively long lead times and the resulting excess inventories by both buyers and sellers.

Thus, there is a need for a system and method that will solve the shortcomings of the prior art described above.

SUMMARY OF THE DISCLOSURE

One aspect of the present invention is directed to a method for managing access to a document by a plurality of users. The document is stored in a memory of a server. The method includes the steps of transmitting a first copy of the document to a computer of a first user and transmitting a second copy of the document to a computer of a second user.

The method further includes receiving a modified first copy of the document from the computer of the first user and storing the modified first copy of the document in the memory under a first document name and receiving a modified second copy of the document from the computer of the second user and storing the modified second copy of the document in the memory under a second document name. In addition, the method includes the step of storing data identifying the modified first copy of the document and data identifying the modified second copy of the document in the memory.

The data stored in the memory includes a first document identifier identifying the modified first copy of the document and a first user identifier identifying the first user. The data also includes a second document identifier identifying the modified second copy of the document and a second user identifier identifying the second user.

In addition, the method includes the steps of authenticating the first user prior to transmitting the first copy of the document to the computer of the first user and authenticating the second user prior to transmitting the second copy of the document to the computer of the second user. The steps of authenticating the first and second users include comparing an identifier provided by each user with data stored in memory that identifies the plurality of users.

Further, the method includes the steps of determining whether the first user already has access to the first copy of the document prior to transmitting the first copy of the document to the computer of the first user, and determining whether the second user already has access to the second copy of the document prior to transmitting the second copy of the document to the computer of the second user. The step of determining whether the first user already has access to the first copy of the document includes comparing a first document identifier and a first user identifier with data stored in a document status table which is stored in the memory. The step of determining whether the second user already has access to the second copy of the document includes comparing a second document identifier and a second user identifier with data stored in the document status table.

The method also includes the steps of determining whether the first user already has access to the first copy of the document prior to receiving the modified first copy of the document and determining whether the second user already has access to the second copy of the document prior to receiving the modified second copy of the document. To determine whether the first user already has access to the first copy of the document, the method includes the step of comparing a first document identifier and a first user identifier with data stored in a document status table which is stored in the memory. To determine whether the second user already has access to the second copy of the document, the method includes comparing a second document identifier and a second user identifier with data stored in the document status table.

Another aspect of the present invention is directed toward a server for managing access to a document by a plurality of users. The server includes a memory, wherein the document is stored, and a processor in communication with the memory. The processor is operative to transmit a first copy of the document to a computer of a first user and transmit a second copy of the document to a computer of a second user.

In addition, the processor is operative to receive a modified first copy of the document from the computer of the first user and store the modified first copy of the document in the memory under a first document name. In addition, the processor is operative to receive a modified second copy of the document from the computer of the second user and store the modified second copy of the document in the memory under a second document name. The processor is also operative to store data identifying the modified first copy of the document and data identifying the modified second copy of the document in the memory.

These and other unique features of the system and method disclosed herein will become more readily apparent from the following detailed description taken in conjunction with the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

So that those of ordinary skill in the art to which the subject invention pertains will more readily understand how to make and use the system and method described herein, embodiments of the invention will be described in detail with reference to the drawings, wherein:

FIG. 1 is a schematic representation of a system configured in accordance with the present invention;

FIG. 2 is a diagram illustrating the creation of a plurality of revised documents made from a single document by a plurality of users;

FIG. 3 is a sample user table identifying users authorized to have access to documents on the computer server;

FIG. 4 is a sample document table identifying documents made available to particular users having access to the computer server;

FIG. 5 is a sample document status table identifying which users presently have a document checked out from a computer server;

FIG. 6 is a sample document revision table identifying the revision of each document that was checked out from the computer server;

FIG. 7 is a flowchart illustrating a process for checking a document out of a computer server, modifying the document, and checking the document back into the computer server;

FIG. 8 is a flowchart more specifically depicting the process illustrated in FIG. 3 associated with checking the document out of a computer server;

FIG. 9 is a flowchart more specifically depicting the process illustrated in FIG. 3 associated with checking the document back into the computer server;

FIG. 10 is a sample seller table similar to the table illustrated in FIG. 3 identifying sellers authorized to have access to documents on the computer server;

FIG. 11 is a sample document table similar to the table illustrated in FIG. 4 identifying documents made available to particular sellers having access to the computer server;

FIG. 12 is a sample document status table similar to the table illustrated in FIG. 5 identifying which suppliers presently have a document checked out of the computer server; and

FIG. 13 is a sample document revision table similar to the table illustrated in FIG. 6 identifying the revision of each document that had been checked out to a seller.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Reference is now made to the accompanying figures for the purpose of describing, in detail, the preferred embodiments of the present invention. The figures and accompanying detailed description are provided as examples of the invention and are not intended to limit the scope of the claims appended hereto.

Throughout the specification that follows, the term "document" is used broadly to identify a named unit of information, which can be stored in a memory, retrieved, edited, and exchanged among computer systems.

Throughout the specification that follows, the term "check-out" is used broadly to identify the action of a user retrieving the document from the computer system. Also "check-in" is used broadly to identify the action of a user to applying changes to the "checked-out" document and storing it back on the computer system. The changed version of any such document will be broadly identified by the term "revision".

Also throughout the specification, the term "data manager" is used broadly to identify software for providing access to particular folders, directories, and documents. The data manager may be either a part of an application program or a routine accessible to an application program. A system administrator may control the operation of the data manager.

In addition, several of the figures used to illustrate attributes of the present invention include "tables" of information. As is well known in the art, tables include a collection of data, usually about a particular topic. Tables store data in "fields" (or columns), and "records" (or rows). In describing embodiments, each record in a table contains the same set of fields, and each field contains the same type of information for each record.

An aspect of the present invention involves the manipulation of information in relational database files. In the described embodiments, Structured Query Language ("SQL") may be utilized to facilitate manipulation of the database files and the information stored therein. SQL is a well-known standard language used to manipulate and retrieve data from a relational database. With SQL, a programmer can, for example, modify a database's structure, query a database for information, and update the contents of a database.

The SQL functions described herein are performed by implementing SQL statements. The SQL statements include, for example, the SELECT statement for retrieving data from a database, the INSERT statement for entering data into a database, and the DELETE statement for deleting data from a database. These statements are referred to below in describing the present invention. The statements can be readily implemented in a number of ways, such as by directly entering the statements, or by embedding them in a computer program and having a computer execute them as a part of one of its routines.

In a preferred embodiment of the present invention, the SQL statements are embedded into an application program. The statements are then available at predetermined instances during execution of the application program. Those skilled in the art will readily understand the various ways in which database manipulation may be accomplished using SQL. In addition, those skilled in the art will recognize that other types of database management systems can be used.

The present invention generally provides a novel and unique system and method for managing documents that may be accessed by a plurality of users. This system and method is capable of allowing, for example, a plurality of users to have access to a copy of an original document from a computer server. The users are able to retrieve and revise the accessed copy of the document, and save the revised copy to the server. The system and method significantly optimizes the utility of documents in a number of settings.

An introductory example illustrating the application of the inventive system and method follows. A board of education for a school district ("board") may use the inventive system and method for requesting vendors to provide quotes for goods and/or services. The board may seek bussing services for transporting students between the schools in its district and the students' homes. In this case, the board would advertise for bids, and then make bid specifications available to potential vendors. To do so, an electronic version of the bid specification would be stored on a computer server that is accessible to the potential vendors through the Internet. The potential vendors would have a time period, i.e., the bidding period, within which to download the specification, fill it in, and transmit the completed bid specification to the board.

During the bidding period, the potential vendors would be able to make a plurality of revisions to the bid specification and store the bid specification on the computer server as the vendors gathered the information (e.g., about student head counts, addresses, and days of service). As each version of the bid specification for a particular vendor is stored to the board's computer server, a new document is created that includes a unique name identifying the new version. The bids are reviewed and selected at a bid opening. The version of the bid specification used at the bid opening by the board would be the last version received, i.e., the last version stored on the computer server by a potential vendor. In this example, all bid specifications would be "opened" at a predetermined time, on a predetermined day, and made available for public review. As is usual in a public bid, the lowest qualified bidder would receive the contract.

As should be evident from the above example, the bidding procedure is substantially streamlined by the use of the present invention in that the latest bid specification from each vendor is available immediately at the time and date needed, i.e., the closing of the bidding period. Other features and advantages will become apparent after the detailed description provided below. Following the detailed description below, a further example illustrating use of the disclosed invention is provided which relates to a request for a quote ("RFQ") for the purchase of goods.

Referring now to the drawings wherein like reference numerals identify similar elements of the present invention, there is illustrated in FIG. 1 a schematic representation of a system 10 in which the present invention may be used. System 10 includes a computer server 12 in communication with client computers 14, 16, and 18 through a computer network 20. Computer network 20 may include any of a number of network systems/subsystems well known to those skilled in the art including, for example, a local area network (LAN), wide area network (WAN), intranet, and the Internet.

Clients 14, 16, and 18 are operated by users 1, 2, and 3, respectively. Users 1, 2, and 3 are referred to collectively as users 19. Although only three clients 14, 16, and 18, are shown, it should be readily apparent that any number of computers can be made to communicate with computer server 12.

Computer server 12 includes a memory 21 which is accessible to each of the users 19. Memory 21 includes a computer program 25, one or more documents 22, and a database 24. The database 24 may include, for example, information regarding the users 19 and documents 22. A data manager 26 controls, either directly or indirectly, the users 19 access to documents 22 stored in the memory 21 utilizing various controls as will be described below.

Generally, users 19 are provided access to particular documents stored in the computer server 12. When a user accesses a document, a copy is created and the user may modify the newly created copy on the user's computer or on the computer server 12. After modifications are made to the document, the user may store the modified version of the document to the memory 21 of the computer server 12. The modified version of the document includes a unique name identifying it as a modified version of the original document. Users may carry out the process described above repeatedly - each time copying the previously modified version of the document. Every copy will include a unique name identifying it as a modified version of the next previously modified version of the document.

Referring to FIG. 2, in a preferred embodiment of the present invention, access to each particular document 1-m is controlled by the data manager 26. As described above, the data manager 26 controls access to particular documents by particular users. To whom a particular document is made available depends on such factors as, for example, the purpose of the document and the criterion set by the system administrator.

For example, restrictions may be applied to document 1 if it is an RFQ for a transportation contract for a public school or if document 1 is an RFQ for the manufacture of subassembly parts. In the case of a bid for a transportation contract for a public school, access to the bid specification may be restricted to school bus companies within a predetermined geographic area. In the case of an RFQ for the manufacture of subassembly parts, access to the RFQ may be restricted to certain subassembly manufacturers that were previously qualified by the requester to manufacture the subassembly parts.

With continued reference to FIG. 2, data manager 26 configured to provide user 1 access to document 1. User 1 is able to download a copy of document 1 into the memory (not shown) of computer 14 (FIG. 1). To download document 1, a check-out procedure is executed in a manner that will be described in detail below. In the preferred embodiments, during the check-out procedure, the modified document is re-named to facilitate the handling of the document. For example, documents can be re-named so that their names include identifying indicia as follows: document 1 (doc1), revision 1 (rev1), user identity (user1), and the name of the document (name). An equivalent truncated version of the new name can be, for example, doc1.rev1.user1.name.

After user 1 has checked out and modified document 1, user 1 can store the revised document in the memory 21 of computer server 12. To store document 1 in memory 21, a check-in procedure is executed in a manner that will be described in detail below. If user 1 attempts to download document 1 a second time, the check-out procedure automatically makes doc1.rev1.user1.name available to user 1 for modification. Again, in the preferred embodiments, the document is renamed, for example, as doc1.rev2.user1.name. After user 1 has modified doc1.rev2.user1.name, user 1 stores it in the memory 21 of computer server 12. Again the check-in procedure is executed.

In a manner similar to that described above, data manager 26 is also configured to provide user 2 with access to document 1. User 2 is able to download a copy of document 1 into the memory (not shown) of computer 16 (FIG. 1). To download document 1, the check-out procedure is executed. In the preferred embodiments, during the check-out procedure, the document is re-named as doc1.rev1.user2.name. After user 2 modifies document 1, user 2 stores it in the memory of computer server 12. To store document 1 in the memory, the check-in procedure is executed. User 2 can make additional modifications to document 1 as illustrated by document revision names doc1.rev2.user2.name and doc1.rev3.user2.name.

Any number of users 19 can be given access to any one of documents 22. In addition, users 19 are allowed to make any number of revisions to those documents to which they have been given access. This is illustrated in FIG. 2, wherein data manager 26 is shown to be configured to provide user n access to document 1, wherein "n" represents a maximum number of users. User n may modify doc 1 in a manner similar to that described above with reference to users 1 and 2. Here, user n has made "q" revisions to document 1.

An embodiment of the invention enables the data manager 26 to be configured to control access to the system 10. For example, the number of users that may access a particular document can be limited. In addition, the number of revisions that a user is able to make can be limited. Furthermore, a user may be permitted to download and alter earlier revisions of documents rather than only the latest version. And still further, a user may designate a particular revision as that which should be considered "final" and ready for review. Some or all of these controls can be provided by the system administrator through the data manager 26. Those skilled in the art will recognize that other controls may be included to facilitate the system administrator's control of the users 19 access to the documents 22 and revised documents 30.

To ensure that the proper document revision is available to the proper user, the users 19 and/or the documents 22 are preferably each identified by a unique identifier code. Identifying each user 19 and document 22 with unique identifiers allows users and documents having the same name to coexist on the system. To efficiently effectuate the process of identifying the users 19 and the documents 22, identifier codes are automatically assigned by the computer program 25 resident in the memory 21 of the computer server 12. The unique identifier codes can be unique alpha-numeric combinations or other appropriate forms of identification. Such identification techniques are well known to those skilled in the art.

Referring to FIGS. 3-6, to facilitate management of the documents 22, four tables are generated and stored in the memory 21 of computer server 12. The four tables are as follows: a user table (FIG. 3), a document table (FIG. 4), a document status table (FIG. 5), and a document revision table (FIG. 6). These tables are created and stored, for example, in the database 24 of the computer server 12 by the data manager 26 (FIG. 1). Of course, those skilled in the art will recognize that other types of tables can be included to facilitate management of the documents 22.

Referring to FIG. 3, the "user table" is illustrated to include two fields of information entitled "user identity" and "user name", identified as 3C1 and 3C2, respectively. There are three exemplary records in the user table shown in FIG. 3. The three records are identified as 3R1, 3R2, and 3R3.

The “user name” field (3C2) includes the name of each user 19. Every user associated with the system has an entry in this table and is identified by a unique identifier. A user's name can be added to the table, for example, by the system administrator, or by the user when the user accesses the computer program 25 on computer server 12. The “user identity” field (3C1) includes a unique identifier of each user 19 that is listed in the user name field (3C2). The three records of the user table identify the following users: user 1, having user identity 1 (3R1); user 2, having user identity 2 (3R2); and user 3, having user identity 3 (3R3)

Referring to FIG. 4, a "document table" is illustrated to include four fields of information entitled "document identity", "document name", "document description", and "document creation date", identified as 4C1, 4C2, 4C3, and 4C4, respectively. There are four exemplary records in the document table illustrated in FIG. 4. The four records are identified as 4R1, 4R2, 4R3, and 4R4.

The "document name" field (4C2) includes the name of each document 22 created and stored in the memory of computer server 12. The "document identity" field (4C1) includes a unique identifier of each document 22 that is listed in the document name field (4C2). In the document table illustrated in FIG. 4, the document name field (4C2) includes document 1, document 2, document 3, and document 4. Each document is identified in the document identity field (4C1) by numeric identifier 1, 2, 3, and 4, respectively. As noted above, documents having the same name can exist as they would have a different document identity.

The other two fields in the document table of FIG. 4, i.e., the "document description" field (4C3) and the "document creation date" field (4C4), provide information regarding the documents. The "document description" field (4C3) includes a brief description of the respective document stored in the memory 21 of computer server 12. The "document creation date" field (4C4) includes the creation date of the respective document.

Referring to FIG. 5, a "document status table" indicates the documents 22 that are presently checked out for modification and to whom they are checked out. After a user has made modifications to a document and checks the document back into the system, the record relating to the particular checked-out document is preferably deleted from the document status table.

The document status table (FIG. 5) includes three fields of information which are entitled "index", "document identity", and "user identity", identified as 5C1, 5C2, and 5C3, respectively. There are three exemplary records in the document status table illustrated in FIG. 5. The three records are identified as 5R1, 5R2, and 5R3.

The "document identity" field (5C2) stores the unique identifier of each document 22 that is listed in the document table (FIG. 4). The "user identity" field (5C3) stores the unique identifier of each user 19 that is listed in the user table (FIG. 3). The "index" field (5C1) includes a unique identifier corresponding to a particular combination of a document identity and a user identity.

The records illustrated in FIG. 5 indicate that there are three documents, i.e., copies of documents in the memory 21 of computer server 12, that are checked out to particular users 19 of the system 10. More specifically, document 2, having document identity number 2, is shown at 5R1 and 5R3 to be checked out to user 2 and user 3 (identified as user identity numbers 2 and 3, respectively). Also, document 1, identified as document identity 1, is shown at 5R2 to be checked out to user 1 (identified as user identity number 1).

Referring to FIG. 6, a "document revision table" includes a record for every document that has been checked out and checked in by any of the users. As described above with respect to FIG. 2, the revision of a document is changed after a user has checked out a document, made changes to the document, and checked in the document. The document revision table includes five fields of information which are entitled "revision identity", "revision number", "document identity", "user identity", and "latest revision", identified as 6C1, 6C2, 6C3, 6C4, and 6C5, respectively. There are six exemplary records in the document revision table illustrated in FIG. 6. The six records are identified as 6R1, 6R2, 6R3, 6R4, 6R5, and 6R6.

The “user identity” field (6C4) includes the unique identifier of each user that has checked a document out and checked the modified document into the system 10. The "document identity" field (6C3) includes the unique identifier of the document that each user has checked out and checked in. The "revision number" field (6C2) includes the revision number given to each document that has been checked out and checked in. The "revision identity" field (6C1) includes a unique identifier corresponding to each record in the document revision table of FIG. 6. The "latest revision" field (6C5), is used to determine the documents representing the latest revisions for each user.

The six exemplary records illustrated in FIG. 6 indicate that several users have checked out several documents a number of times. More specifically, as shown by records 6R1, 6R3, and 6R5, user 1 (user identity number 1) is shown to have checked out and checked in document 1 (document identity number 1) three times. The revision identity field (6C1) identifies the three revisions as revision identities 1, 3, and 5 for revision numbers 1, 2, and 3, respectively.

In a similar manner, record 6R4 shows that user 2 (user identity number 2) has checked out and checked in document 2 (document identity number 2) once. The revision identity field (6C1) identifies the revision as revision identity 4 for revision number 1. Also, records 6R2 and 6R6 shows that user 3 (user identity number 3) checked out and checked in document 2 (document identity number 2) twice. The revision identity field (6C1) identifies the two revisions as revision identities 2 and 6 for revision numbers 1 and 2, respectively.

The latest revision of each document for each user is identified by an asterisk in the latest revision field (6C5). There is no entry in the document revision table (FIG. 6) for document 4 (document identity number 4). This is because document 4 was never checked out by any of the users.

It is notable in the description of FIG. 6 above that each user is shown to have access to a particular document. More specifically, user 1 (user identity 1) is shown to have access to document 1 (document identity 1), user 2 (user identity 2) is shown to have access to document 2 (document identity 2), and user 3 (user identity 3) is shown to have access to document 2 (document identity 2). As described above, an embodiment of the present invention enables the system administrator to control the system through the data manager 26 in a variety of ways. For example, the system administrator may permit each user to have access to all of the documents, all revisions of a particular document, only the latest revision of a particular document, etc.

As an example of the present invention's flexibility, the system administrator may permit user 1 to have access to documents 1-4 found in the document table (FIG. 4). At the same time, the system administrator may permit user 2 to have access to only documents 1 and 2 found in the document table. Further, the system administrator may permit both user 1 and user 2 to have access to all revisions of document 1 but only the latest revision of each of the remaining documents to which they have access. It is readily apparent that the system administrator has a great degree of flexibility in deciding how to configure access to the documents 22 and revisions 30 thereto. How the system is configured will depend largely on the purpose of the document to which access is being provided.

Referring to FIGS. 1 and 7, documents are checked out of computer server 12, modified, and checked back into the computer server 12 as follows. At step 60, a user starts the procedure by logging onto the computer server 12. At step 61, the user gains access to the computer program 25 and attempts to download a document into his computer. In attempting to do so, the computer program 25 first identifies the user and verifies that the user has the right to access the document. If the computer program 25 determines that the user does not have the right to access the document, the user will be denied access thereto. If the computer program 25 determines that the user has the right to access the document, the computer program 25 carries out a check-out procedure. This procedure is described in more detail below and illustrated in FIG. 8.

At step 62, after the computer program 25 determines the user has the right to access the document and allows the user to check out the document, the user downloads the document to his computer. The user is able to work with the document, e.g., make modifications to it and store it, on his own computer in a manner well known to those skilled in the art. At such time when the user is working with the document, the user can maintain or terminate communication with computer server 12.

At step 63, after the user completes the modifications to the document, the user attempts to store the document on computer server 12. If communication with computer server 12 was terminated after checking out the document, the user must reestablish communication with computer server 12. When attempting to store the document on computer server 12, the computer program 25 again identifies the user and verifies that the user has the right to store the document. If the computer program 25 determines that the user does not have the right to store the document, the user will not be permitted to store the document. If the computer program 25 determines that the user has the right to store the document, the computer program 25 performs a check-in procedure. This procedure is described in more detail below and is illustrated in FIG. 9.

After the document has been modified and checked into the computer server 12, the user terminates communication with the computer server 12. The procedure ends at step 64. An example of the transformation of the document described above from an original document to a revised document is illustrated in FIG. 2. For example, user 1 is illustrated to have accessed document 1 and generated a revised document that was named as doc1.rev1.user1.name. Thereafter, user 1 is shown to have accessed document doc1.rev1.user1.name and generated a revised document that was named doc1.rev2.user1.name.

Referring now to FIG. 8, the procedure for checking a document out from the computer server 12, described broadly above with respect to step 61 of FIG. 7, is described in more detail below. In describing the procedure, reference will be made to the tables illustrated in FIGS. 3-6.

At step 70, the procedure is started for checking out a document from the memory 21 of the computer server 12. At step 71, after the user is identified by the computer program 25, for example, by entering a user identity, the computer program 25 initiates the document check-out procedure. The user's identity is temporarily stored in the memory 21 of the computer server 12. The computer program 25 uses the user identity in an attempt to authenticate the user as a valid user.

The authentication process may be accomplished by executing a SELECT statement to select users identities from the user identity field 3C1 of the user table (FIG. 3). If the identity of the user entered by the user, does not correspond to any of the user identities listed in the user table (FIG. 3), an error message will result and, at step 72, the user will be denied access to any documents 22 stored in the memory 21 of the computer server 12. If, however, the identity of the user does correspond to a user identity listed in the user table (FIG. 3), the user will be requested, at step 73, to enter a document name to which the user desires access.

At step 73, the user enters the name of the document to which the user desires access. At step 74, the computer program 25 executes a SELECT statement to select all of the records (5R1-5R3) in the document status table (FIG. 5). The computer program 25 then makes a comparison between the records in the document status table (FIG. 5) and the combination of the user identity (FIG. 3) and the document identity (FIG. 4) of the document that the user desires to access.

If, at step 74, it is determined that there is a match between a record in the document status table (FIG. 5) and the combination of the user identity (FIG. 3) and the document identity (FIG. 4), an error message will result indicating that the document is already checked out by the user and, at step 75, the user will be denied access to the desired document.

For example, if user 1 desires to check out document 1 from the computer server 12, the computer program 25 will determine if there is a match between a record in the document status table (FIG. 5) and the combination of the user identity, i.e., user identity 1 (FIG. 3), and the document identity, i.e., document identity 1 (FIG. 4). In this example, the record identified as index 1 (5R2) in the document status table (FIG. 5) matches the combination. Consequentially, an error message will result indicating that document 1 is already checked out by user 1 and that he cannot access document 1 until after it is checked back into the computer server 12.

If, at step 74, it is instead determined that there is no match between a record in the document status table (FIG. 5), and the combination of the user identity (FIG. 3) and the document identity (FIG. 4), the computer program 25 retrieves either the original copy of the document, i.e., a copy of the document as originally stored by the system administrator, or the last version of the document as revised by the user.

In order to determine which document to retrieve, the computer program 25 executes a SELECT statement to select the records representing the latest document revisions from the document revision table (FIG. 6), i.e., records 6R4, 6R5, and 6R6. Thereafter, computer program 25 makes a comparison between the records representing the latest document revisions (FIG. 6) and the combination of the users identity (FIG. 3) and the document identity of the document that the user desires to access (FIG. 4).

If, at step 76, it is determined that there is no match between the records representing the latest document revisions (FIG. 6), and the combination of the users identity (FIG. 3) and the document identity (FIG. 4), the computer program 25 retrieves, at step 77, an original copy of the document from the memory 21 of the computer server 12. If, however, it is determined that there is a match between the records representing the latest document revisions (FIG. 6) and the combination of the users identity (FIG. 3) and the document identity (FIG. 4), the computer program 25 instead retrieves, at step 78, the latest revision of the document for the user.

For example, if user 1 desires to check out document 4 from the computer server 12, the computer program 25 will determine if there is a match between the records in the document revision table (FIG. 6) and the combination of the user identity, i.e., user identity 1 (FIG. 3), and the document identity, i.e., document identity 4 (FIG. 4). In this example, it is determined that there is no match, therefore, the computer program 25 retrieves an original copy of document 4 from the memory 21 of the computer server 12.

On the other hand, if user 1 desires to check out document 1 from the computer server 12, the computer program 25 will determine if there is a match between the records in the document revision table (FIG. 6) and the combination of the user identity, i.e., user identity 1 (FIG. 3), and the document identity, i.e., document identity 1 (FIG. 4). In this case, the record identified as revision identity 5 (6R5) in the document revision table (FIG. 6) matches the combination. Consequentially, the computer program 25 retrieves the latest revision (revision identity 5 having revision number 3) of document 1 for user 1.

After a document has been selected from the memory 21 of computer 25 by following step 76 and step 77 to retrieve an original copy, or step 76 and step 78 to retrieve a latest document revision as described above, the document status table (FIG. 5) and the document revision table (FIG. 6) are updated. At step 79, the document status table (FIG. 5) is updated by the computer program 25 executing an INSERT statement to insert a record into the table (FIG. 5) including the identity of the document being accessed and the user identity. An index identifier is also inserted to identify the record.

In a similar manner, at step 80, the document revision table (FIG. 6) is updated by the computer program 25 executing an INSERT statement to insert a record into the table (FIG. 6) including the document identity and the user identity. In addition, the next revision number is assigned to the record. Also, the asterisk will be shifted to the latest document revision. A revision identity is also inserted to identify the record.

At step 81, if no error message was received by the user at steps 71 and 72, or steps 74 and 75, the user has successfully checked the desired document out and is able to modify and store the document on the user's computer as described above for step 62 (FIG. 7).

Referring now to FIG. 9, an embodiment of the procedure for checking a document into the computer server 12, described broadly above with respect to step 63 of FIG. 7, is described in more detail below. In describing the procedure, reference will be made to the tables illustrated in FIGS. 3-6.

At step 100, after the user has made modifications to the document, the user starts the check-in procedure. At step 101, the user initiates the document check-in procedure by gaining access to the computer program 25 on the computer server 12 and identifying himself/herself to the computer program 25, for example, by entering a user identity. The user's identity is temporarily stored in the memory 21 of the computer server 12.

The computer program 25 attempts to authenticate the user as a valid user. The authentication process may be accomplished by executing a SELECT statement to select users identities from the user identity field (3C1) in the user table (FIG. 3). If the identity of the user, for example, the user identity that was entered by the user, does not correspond to any of the user identities listed in the user table (FIG. 3), an error message will result and, at step 102, the check-in procedure will be terminated. If, however, the identity of the user does correspond to a user identity from the user identity field (3C1) in the user table (FIG. 3), the user will, at step 104, be requested to enter the name of the document for which the user desires to check in, i.e., store in the memory 21 of computer server 12.

At step 103, the user enters the name of the document the user desires to check in. Thereafter, the computer program 25 executes a SELECT statement to select all of the records (5R1-5R3) in the document status table (FIG. 5). The computer program 25 then makes a comparison between the records in the document status table (FIG. 5), and the combination of the users identity and the document identity of the document that the user desires to check in.

If, at step 104, it is determined that there is not a match between a record in the document status table (FIG. 5), and the combination of the users identity and the document identity, an error message will result indicating that the document was not earlier checked out from the computer server 12 and, at step 105, the user will be not be permitted to check the document into the memory 21 of computer server 12.

If, at step 104, it is instead determined that there is a match between a record in the document status table (FIG. 5), and the combination of the users identity and the document identity, the computer program 25 stores the modified document into the memory 21 of computer server 12. Thereafter, the computer software 25 differentiates between a document that is a first revision of an original document, i.e., modified for the first time, or a document that is a revision subsequent to a first revision of an original document. A first revision of an original document will be stored as a new document in the memory 21 of computer server 12. A revision subsequent to a first revision of an original document can be written over a previous revision of the document or stored as a new document. Of course, a previous revision that is overwritten will be unretrievable. The system administrator, through the data manager 26, can establish whether revisions subsequent to a first revision of an original document should be overwritten or not.

After the computer program 25 has stored the modified document into the memory 21 of computer server 12, the document status table (FIG. 5) is updated. In addition, as an alternative to having done so during the check-out procedure, the document revision table (FIG. 6) is updated.

At step 107, the document status table (FIG. 5) is updated by the computer program 25 executing a DELETE statement to remove the record of the document that is being checked in, the record that includes the index number, document identity, and the user identity of the document being checked in.

At step 108, to update the document revision table during the check-in procedure, as an alternative to having done so during the check-out procedure, the computer program 25 executes a SELECT statement to select from the document revision table (FIG. 6) the record representing the latest revision of the document to be checked in. If a record is found for the document, a new record is created including the next revision identity and revision number for the document. Also, the asterisk is also shifted to the latest document revision.

If a record is not found for the document, a new record is created including the next revision identity and a first revision number. To update the document revision table (FIG. 6), the computer program 25 executes an INSERT statement to insert the new record into the table.

At step 109, if no error message was received by the user at steps 101 and 102, or steps 104 and 105, the user has successfully checked the revised document into the memory 21 of the computer server 12.

When documents are stored to the memory 21 of the computer server 12, they can be identified with unique names. For example, the names may include information that describes a document as follows: a) document identity (FIG. 4); b) revision identity (FIG. 6); c) user identity (FIG. 3); and c) document name (FIG. 4). Using a unique name having the format described above can provide certain advantages during the checking-out (FIG. 8) and checking-in (FIG. 9) procedures described above. For example, during the checking-in procedure, at step 104 (FIG. 9), instead of the computer program 25 executing a SELECT statement to select the relevant document identity and user identity in the document revision table (FIG. 6), the computer program 25 can instead directly evaluate the unique name of the document to be checked in.

To further illustrate the present invention, an example employing the above described system and method is provided. To facilitate the description, reference should be made to system 10 in FIG. 1. In the example, a manufacturer desires to purchase various materials to support the manufacturing of light bulbs. In an effort to expeditiously procure the various materials, the manufacturer creates an electronic bid board on computer server 12. The bid board in this case might be, for example, a browser-based application that is resident on the manufacturer's computer server 12. The bid board acts as a common location wherein, for example, various vendors of materials can log on to respond to RFQs that are stored on the bid board. The manufacturer may initiate the quoting process by storing an RFQ in the form of an electronic document on the bid board.

Further, suppose that the bulb manufacturer requires to purchase filaments from a vendor to support the production of light bulbs. Hereinafter, the light bulb manufacturer is referred to as the buyer while the filament vendors are referred to as the sellers. To obtain the needed filaments, the buyer generates an electronic RFQ. In the RFQ, a seller is prompted to provide, for example, a list of prior customers, manufacturing standards which are met by the seller, price breaks for various quantities of filaments, and guaranteed delivery dates.

To obtain the quotes from sellers, the buyer informs selected sellers when the RFQ will be stored on the bid board. For example, the buyer may contact sellers by e-mail. For this example there are two sellers, referred to herein as seller-A and seller-B, that are contacted by the buyer to bid for the contract to supply filaments to the buyer.

If a seller wants to respond to the RFQ, the seller checks out the RFQ document named "RFQ for filaments 2204" (which will be stored on the bid board) from the buyer's computer server 12, modify the document by filling in the required fields, and check the modified document back into the computer server 12. If either seller later desires to modify a previously modified document, the seller can check the document out again and make further modifications. Thereafter, the seller will check the document back into the computer server 12. After a time period as specified by the buyer, the buyer will review the last-modified documents received from the sellers to determine which seller is to receive the contract to supply the filaments.

To initiate the bidding process, the buyer adds the above described RFQ document to the electronic bid board on computer server 12. In addition, the buyer executes an INSERT statement to enter the names of the sellers who are authorized to bid on the contract to supply the filaments. Referring to FIG. 10, a seller table (analogous to the user table of FIG. 3 above) is illustrated and may include the names of the sellers ("seller-A" and "seller-B") and a unique identifier for each seller ("1" and "2", respectively). With their names and identity in the seller table, both seller-A and seller-B would be approved as authorized sellers to bid on the contract.

Furthermore, to initiate the bidding process, the buyer executes an INSERT statement to make an entry into a document table residing in the memory of computer server 12. The document table, illustrated in FIG. 11, includes the document identity ("1"), document name ("RFQ for filaments 2204"), document description ("Request for quote for the procurement of filaments for project number 2204"), and document creation date ("04-24-98") of the RFQ.

To effectuate the check-out process, seller-A and seller-B must initially identify themselves to the computer program 25, for example, by entering user identities previously assigned to them by the buyer. The computer user identities may be, for example, the seller identity assigned by the buyer in the seller table (FIG. 10). The computer program 25 attempts to verify that the sellers are valid by comparing their user identities with the list of approved vendors in seller table (FIG. 10).

After a seller is approved as a valid user, the computer program 25 may then prompt the seller to enter the name of the document that it desires to access. After the document name, in this example, "RFQ for filaments 2204", is entered, the computer program 25 executes a SELECT statement to select document identities from the document table (FIG.

11). In addition, the computer program 25 executes a SELECT statement to select all of the records in a document status table, for example, such as that discussed above and illustrated in FIG. 5.

The computer program 25 then makes a comparison between the records in the document status table and the combination of the seller identity (FIG. 10) and the document identity of the document that the seller desires to access (FIG. 11). In the instant example, where the sellers are attempting to access the document for the first time, there would not be an entry in the document status table for the combination of seller-A's or seller-B's identity (FIG. 10) and the document identity (FIG. 11), therefore, the computer program 25 would retrieve an original copy of the document entitled "RFQ for filaments 2204".

Referring to FIGS. 12 and 13, after a seller has accessed the document from computer server 12, the document status table (FIG. 12) is updated. In the present example, after seller-A has retrieved the document entitled "RFQ for filaments 2204", the document status table (FIG. 12) is updated by the computer program 25 executing an INSERT statement to insert a record into the table including the document identity ("1") and seller-A's identity ("1"). The record is also identified by an index number ("1"). Similarly, after seller-B has retrieved the document entitled "RFQ for filaments 2204", the document status document status table (FIG. 12) is updated to include a record including the document identity ("1") and seller-B's identity ("2"). The record is also identified by an index number ("2"). The resulting document status table is illustrated in FIG. 12.

After the check-out procedure is completed, the computer program 25 retrieves the document and makes its contents available to the requesting seller. The sellers can then retrieve the document for modification, i.e., completion of the bid specification. Also, the document can be renamed by the computer server 12 to more particularly describe the document. As described above, the name may include, for example, a combination of identifying information as follows: a) document identity (FIG. 11); b) revision identity (FIG. 13); c) seller identity (FIG. 10); and d) document name (FIG. 11). Using the afore described document storage name format, the document accessed by seller-A in the above example may have the name "1.1.1.RFQ for filaments 2204" and the document accessed by seller-B in the above example may have the name "1.1.2.RFQ for filaments 2204".

After the sellers have made modifications to the document, each seller will seek to check the document into computer server 12. The sellers can initiate the check-in procedure (FIG. 9) by gaining access to the computer program 25 on the computer server 12 and identifying himself/herself to the computer program 25, for example, by entering a user identity as was done during the check-out procedure described above.

The computer program 25 attempts to authenticate the seller. The authentication process is accomplished by executing a SELECT statement to select sellers identities from the seller table (FIG. 10). As noted above, for the present example, both seller-A and seller-B are approved as authorized sellers.

When a seller is authenticated as a valid seller, the seller then transmits to the computer program 25 the name of the document to be checked into the computer server 12. Thereafter, the computer program 25 executes a SELECT statement to select all of the records in the document status table (FIG. 12) and compares the records in the table with the identification of the seller (FIG. 10) and the document to be checked in (FIG. 11). In the present example, since seller-A and seller-B were identified as having checked out document identity 1 ("RFQ for filaments 2204"), they will now be permitted to check in the modified version of the document.

It should be noted that using the modified name format, e.g., "1.1.1.RFQ for filaments 2204" for seller-A and "1.1.2.RFQ for filaments 2204" for seller-B, can facilitate a more convenient check-in procedure. For example, instead of the computer program 25 having to receive two entries from each seller during check in, i.e., the seller's name and the document's unique name, the computer program can receive the name alone. In such case, the computer can evaluate the name since it conveniently includes all of the information necessary for authenticating the supplier's identity, document identity, and document revision level.

After it has been established that a seller previously checked out a document and is now permitted to check in the modified document, a new record must be created to identify the new revision of the document. For example, after seller-A has checked in modified document identity 1 ("RFQ for filaments 2204"), document revision table (FIG. 13) is updated by the computer program 25 executing an INSERT statement to insert a record into the table to include the document identity ("1") and the seller's identity ("1"). In addition, the next revision number for seller-A is assigned to the record. Since seller-A modified a copy of the original document, the document will be identified as a first revision ("1"). Also, the record is also identified by the next available revision identity ("1").

With continued reference to FIG. 13, similarly, after seller-B has checked in modified document identity 1 ("RFQ for filaments 2204"), document revision table (FIG. 13) is updated by the computer program 25 executing an INSERT statement to insert a record into the table to include the document identity ("1") and the seller identity ("2"). In addition, the next revision number for seller-B is assigned to the record. Since seller-B also modified a copy of the original document, the document will be identified as a first revision ("1"). Also, the record is identified by the next available revision identity ("2").

After a document has been checked in, the computer program 25 removes from document status table (FIG. 12) the records indicating that seller-A and seller-B have document identity 1 ("RFQ for filaments 2204") checked out. To do so, computer program

25 executes a DELETE statement to delete the record from the document status table (FIG. 12) identified as index "1". Similarly, computer program 25 executes a DELETE statement to delete the record from the document status table (FIG. 12) identified as index "2".

As noted above, if either seller later desires to modify a previously modified document, the seller can do so following the checking-out procedure (FIG. 8) and checking-in procedure (FIG. 9) as described above. After the time period as specified by the buyer, the buyer will review the last-modified documents received from the seller-A and seller-B to determine which seller is to receive the contract to supply the filaments.

Thus, by utilizing the system and method of the present invention, the manufacturer has acquired the quotes for the desired filaments in an efficient and expeditious manner. More specifically, the costs, amount of time spent, and trouble normally associated with the process has been substantially reduced.

Those skilled in the art will recognize that although the descriptions and illustrations above discuss checking in and checking out documents between computers via a computer network, the same procedures can be carried out directly on the computer server 12. That is, computer server 12 can be configured to enable approved individuals to directly utilize its keyboard, monitor, mouse, etc. to check out a document, make modifications thereto, and thereafter check in the modified document.

It is clear from the foregoing disclosure that the present system and method for managing documents that are made available to a plurality of users provides an advancement in the art of document management. The system and method allows sellers to respond electronically to a buyers requests for quotes. In addition, the system and method allows sellers to store their responses in a computer server, thereby the responses are readily accessible to both the buyer and seller. Furthermore, the system and method provides a computer server that is capable of storing a plurality of versions/revisions of a document. The system and method also provides access to each seller's specific revisions to the document only by the specific seller that prepared the revisions and to the buyer.

While the system and method disclosed has been described with respect to various specific embodiments, those skilled in the art will readily appreciate that various modifications, changes, and enhancements may be made thereto without departing from the spirit and scope of the invention as defined by the appended claims.

What is claimed is:

1. A method for managing access to a document by a plurality of users, wherein the document is stored in a memory of a server, and wherein the method comprises:

(a) transmitting a first copy of the document to a computer of a first user;

(b) transmitting a second copy of the document to a computer of a second user;

(c) receiving a modified first copy of the document from the computer of the first user and storing the modified first copy of the document in the memory under a first document name;

(d) receiving a modified second copy of the document from the computer of the second user and storing the modified second copy of the document in the memory under a second document name; and

(e) storing data identifying the modified first copy of the document and data identifying the modified second copy of the document in the memory.

2. A method as recited in Claim 1, wherein the data stored in step (e) comprises:

(a) a first document identifier identifying the modified first copy of the document and a first user identifier identifying the first user; and

(b) a second document identifier identifying the modified second copy of the document and a second user identifier identifying the second user.

3. A method as recited in Claim 1, further comprising the steps of authenticating the first user prior to transmitting the first copy of the document to the computer of the first user and authenticating the second user prior to transmitting the second copy of the document to the computer of the second user.

4. A method as recited in Claim 3, wherein the steps of authenticating the first and second users comprises comparing an identifier provided by each user with data stored in the memory that identifies the plurality of users.

5. A method as recited in Claim 1, further comprising the steps of:

(a) determining whether the first user already has access to the first copy of the document prior to transmitting the first copy of the document to the computer of the first user; and

(b) determining whether the second user already has access to the second copy of the document prior to transmitting the second copy of the document to the computer of the second user.

6. A method as recited in Claim 5, wherein the steps of:

(a) determining whether the first user already has access to the first copy of the document comprises comparing a first document identifier and a first user identifier with data stored in a document status table which is stored in the memory; and

(b) determining whether the second user already has access to the second copy of the document comprises comparing a second document identifier and a second user identifier with data stored in the document status table.

7. A method as recited in Claim 1, further comprising the steps of:

(a) determining whether the first user already has access to the first copy of the document prior to receiving the modified first copy of the document; and

(b) determining whether the second user already has access to the second copy of the document prior to receiving the modified second copy of the document.

8. A method as recited in Claim 7, wherein the steps of:

(a) determining whether the first user already has access to the first copy of the document comprises comparing a first document identifier and a first user identifier with data stored in a document status table which is stored in the memory; and

(b) determining whether the second user already has access to the second copy of the document comprises comparing a second document identifier and a second user identifier with data stored in the document status table.

9. A server for managing access to a document by a plurality of users, comprising:

(a) a memory storing the document; and

(b) a processor in communication with the memory, wherein the processor is operative to

(i) transmit a first copy of the document to a computer of a first user;

(ii) transmit a second copy of the document to a computer of a second user;

(iii) receive a modified first copy of the document from the computer of the first user and store the modified first copy of the document in the memory under a first document name;

(iv) receive a modified second copy of the document from the computer of the second user and store the modified second copy of the document in the memory under a second unique document name; and

(v) store data identifying the modified first copy of the document and data identifying the modified second copy of the document in the memory.

10. A server as recited in Claim 9, wherein the data stored the memory comprises

(a) a first document identifier identifying the modified first copy of the document and a first user identifier identifying the first user; and

(b) a second document identifier identifying the modified second copy of the document and a second user identifier identifying the second user.

11. A server as recited in Claim 9, wherein the processor is further operative to authenticate the first user prior to the processor transmitting the first copy of the document to the computer of the first user and operative to authenticate the second user prior to the processor transmitting the second copy of the document to the computer of the second user.

12. A server as recited in Claim 11, wherein to authenticate the first and second users the processor is further operative to compare an identifier provided by each user with data stored in memory that identifies the plurality of users.

13. A server as recited in Claim 9, wherein the processor is further operative to

(a) determine whether the first user already has access to the first copy of the document prior to the processor transmitting the first copy of the document to the computer of the first user; and

(b) determine whether the second user already has access to the second copy of the document prior to the processor transmitting the second copy of the document to the computer of the second user.

14. A server as recited in Claim 13, wherein to determine whether the first user already has access to the first copy of the document, the processor compares a first document identifier and a first user identifier with data stored in a document status table which is stored in the memory.

15. A server as recited in Claim 13, wherein to determine whether the second user already has access to the second copy of the document, the processor compares a second document identifier and a second user identifier with data stored in a document status table which is stored in the memory.

16. A server as recited in Claim 9, wherein the processor is further operative to

(a) determine whether the first user already has access to the first copy of the document prior to receiving the modified first copy of the document; and

(b) determine whether the second user already has access to the second copy of the document prior to receiving the modified second copy of the document.

17. A server as recited in Claim 16, wherein to determine whether the first user already has access to the first copy of the document, the processor compares a first document identifier and a first user identifier with data stored in a document status table which is stored in the memory.

18. A server as recited in Claim 16, wherein to determine whether the second user already has access to the second copy of the document, the processor compares a second document identifier and a second user identifier with data stored in a document status table which is stored in the memory.

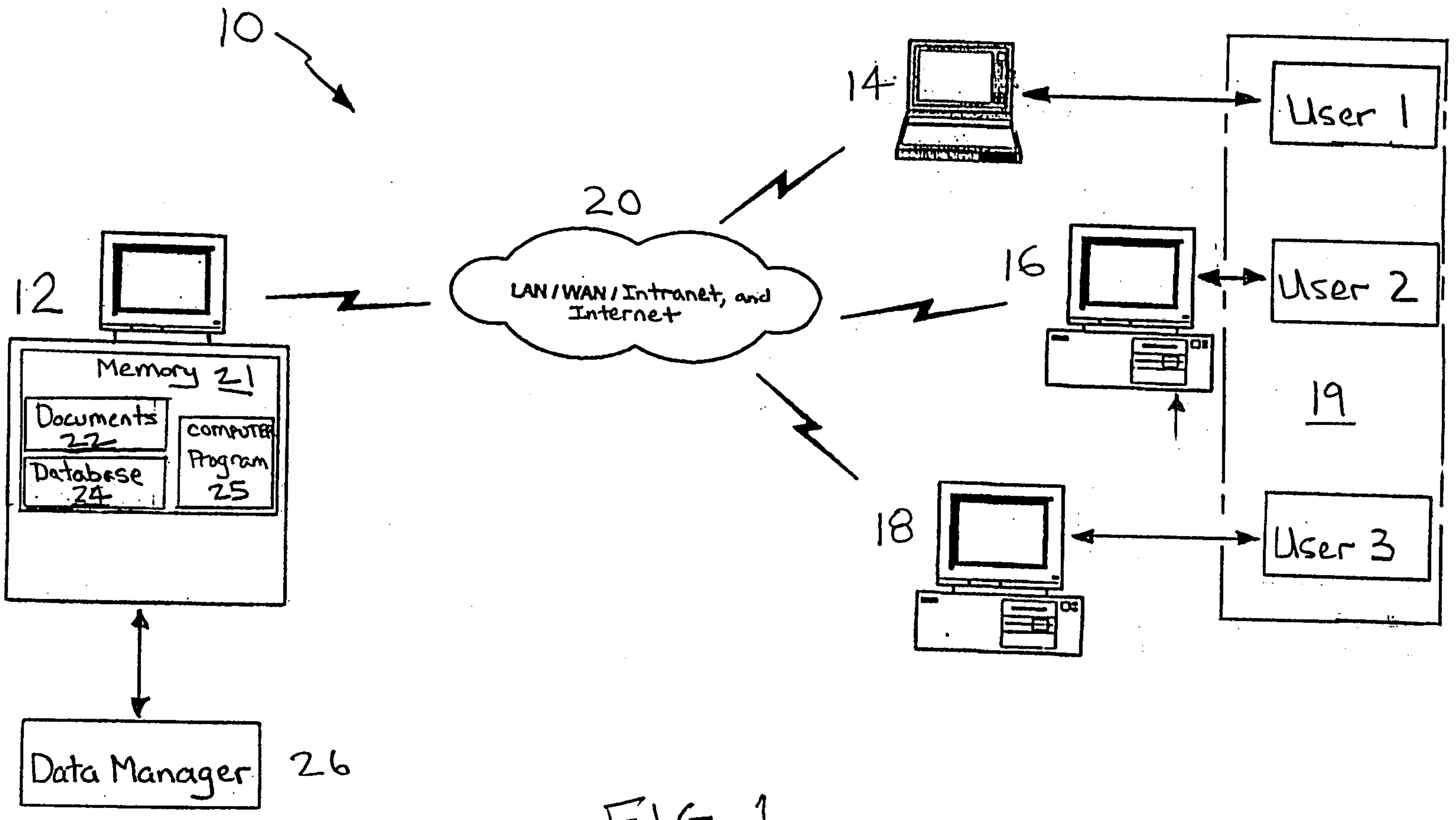
10
12
Memory 21
Documents 22
Database 24
COMPUTER Program 25
Data Manager
26
20
LAN/WAN/Intranet, and Internet
14
16
18
User 1
User 2
User 3
19

FIG. 1

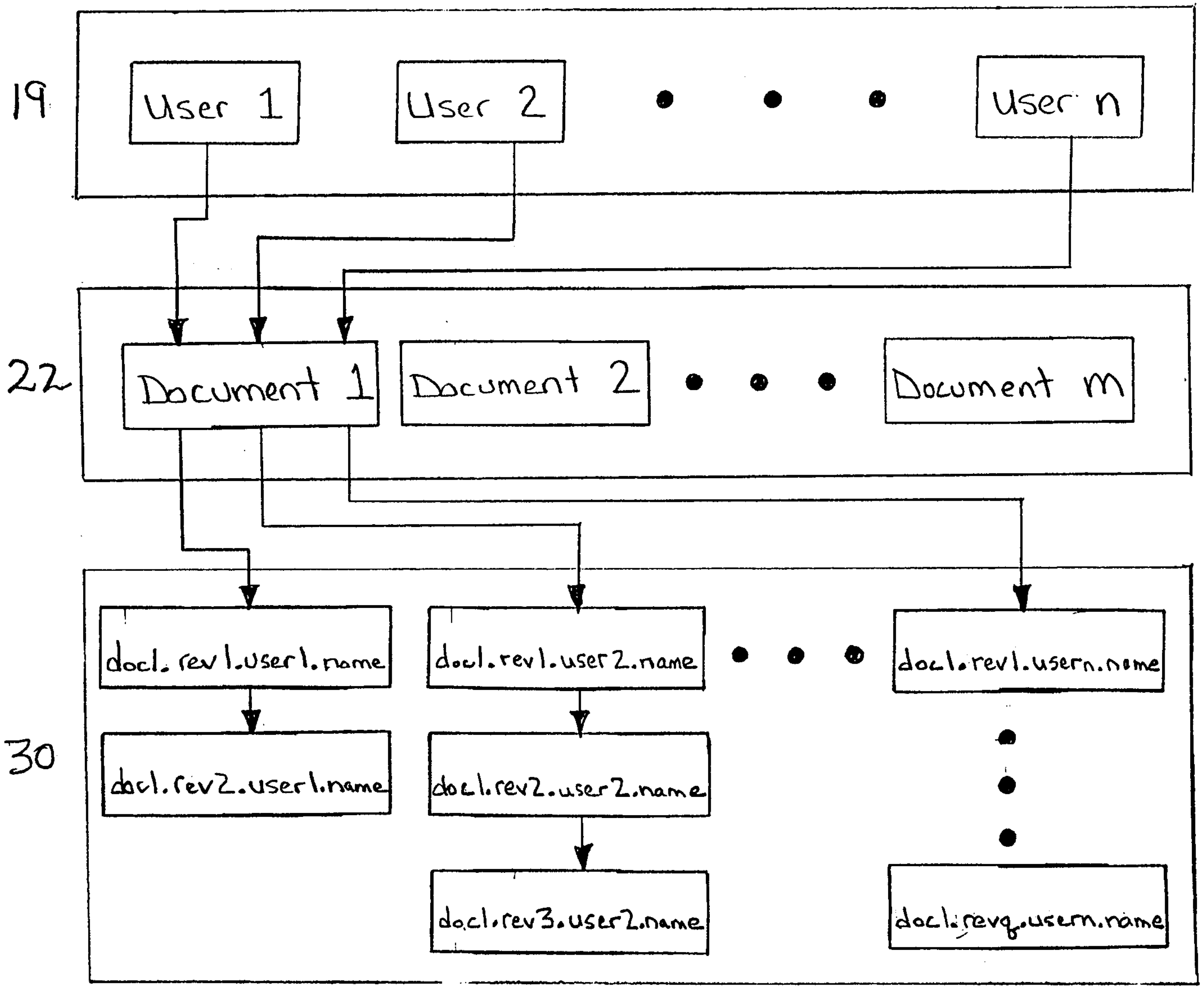
19
User 1
User 2
User n
22
Document 1
Document 2
Document m
30
docl. revl.userl.name
docl.rev2.userl.name
docl.revl.user2.name
docl.rev2.user2.name
docl.rev3.user2.name
docl.revl.usern.name
docl.revq.usern.name

FIG. 2

	3C1	3C2
	User Identity	User Name
3R1	1	user 1
3R2	2	user 2
3R3	3	user 3

USER TABLE
FIG. 3

	4C1	4C2	4C3	4C4
	Document Identity	Document Name	Document Description	Document Creation Date
4R1	1	Document 1	This forms the white paper for the project.	04-24-98
4R2	2	Document 2	Terms and conditions for all clients.	03-30-98
4R3	3	Document 3	Project Guidelines	10-19-97
4R4	4	Document 4	Standards	09-12-97

DOCUMENT TABLE
FIG. 4

	5C1	5C2	5C3
	Index	Document Identity	User Identity
5R1	1	2	2
5R2	2	1	1
5R3	3	2	3

DOCUMENT STATUS TABLE
FIG. 5

	6C1	6C2	6C3	6C4	6C5
	Revision Identity	Revision Number	Document Identity	User Identity	Latest Revision
6R1	1	1	1	1	
6R2	2	1	2	3	
6R3	3	2	1	1	
6R4	4	1	2	2	*
6R5	5	3	1	1	*
6R6	6	2	2	3	*

DOCUMENT REVISION TABLE

FIG. 6

Seller Identity	Seller Name
1	Seller-A
2	Seller-B

SELLER TABLE

FIG. 10

Document Identity	Document Name	Document Description	Document Creation Date
1	RFQ for filaments 2204	Request for quote for the procurement of filaments for project number 2204.	04-24-98

DOCUMENT TABLE

FIG. 11

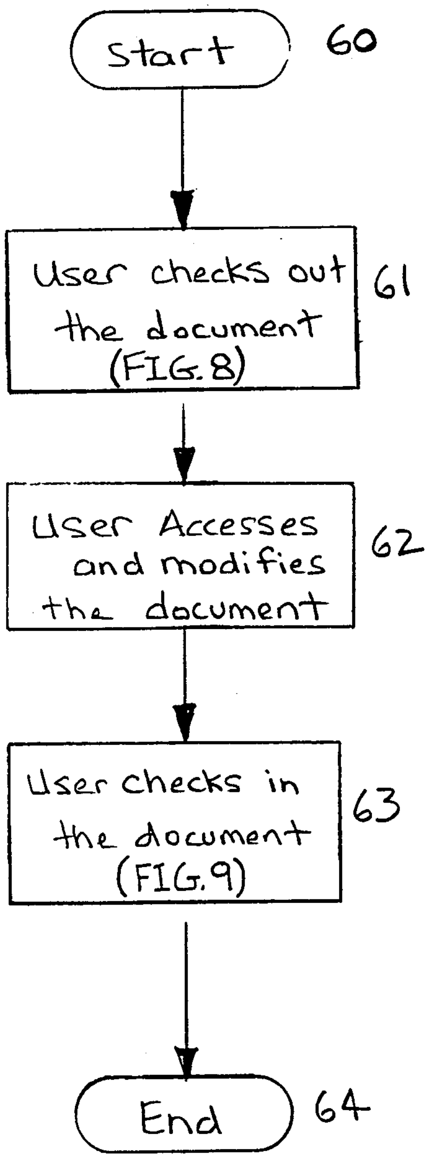
Start
60
User checks out the document (FIG. 8)
61
User Accesses and modifies the document
62
User checks in the document (FIG. 9)
63
End
64

FIG. 7

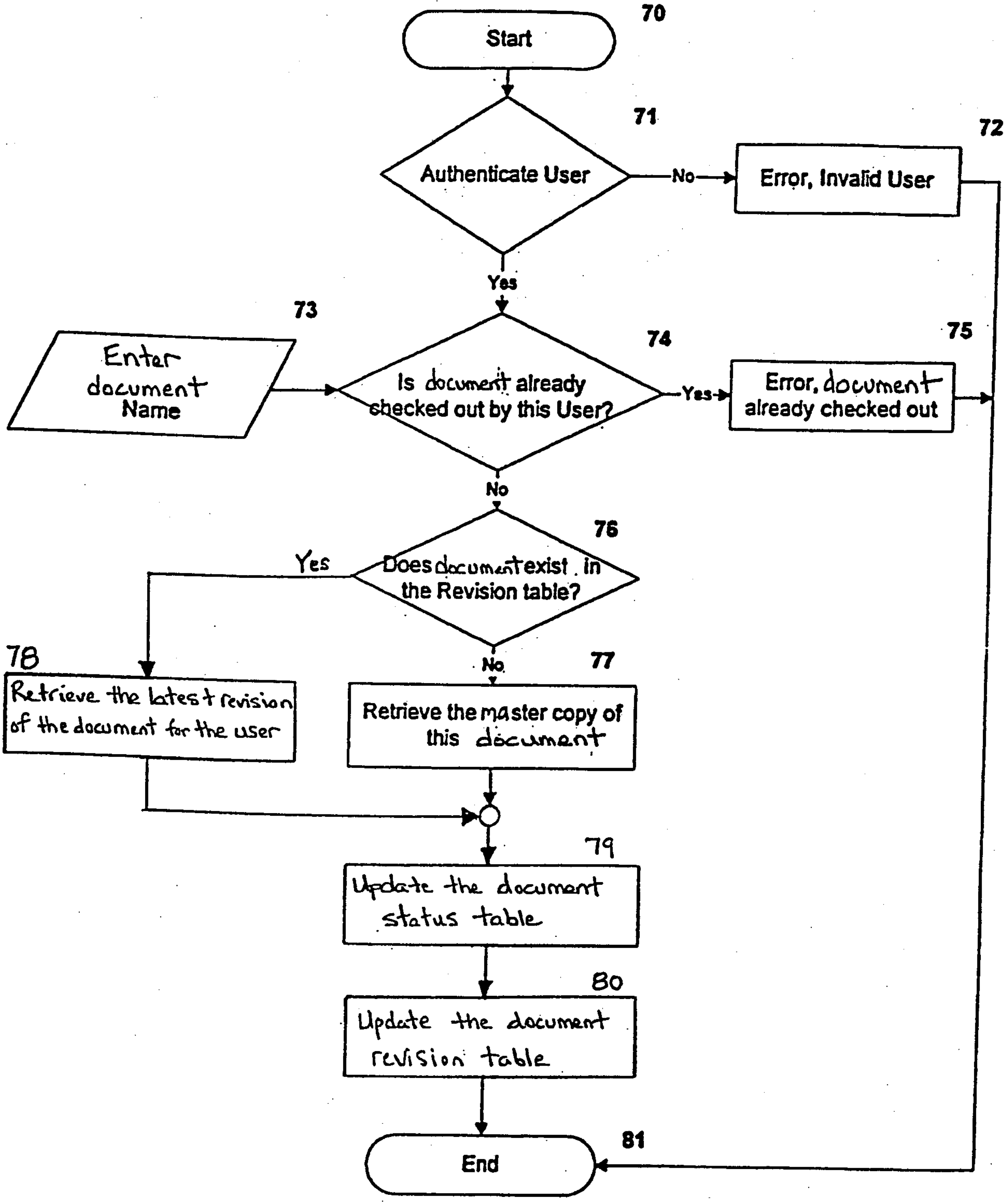
70
Start
71
Authenticate User
No
72
Error, Invalid User
Yes
73
Enter document Name
74
Is document already checked out by this User?
Yes
75
Error, document already checked out
No
76
Does document exist in the Revision table?
Yes
78
Retrieve the latest revision of the document for the user
No
77
Retrieve the master copy of this document
79
Update the document status table
80
Update the document revision table
81
End

FIG. 8

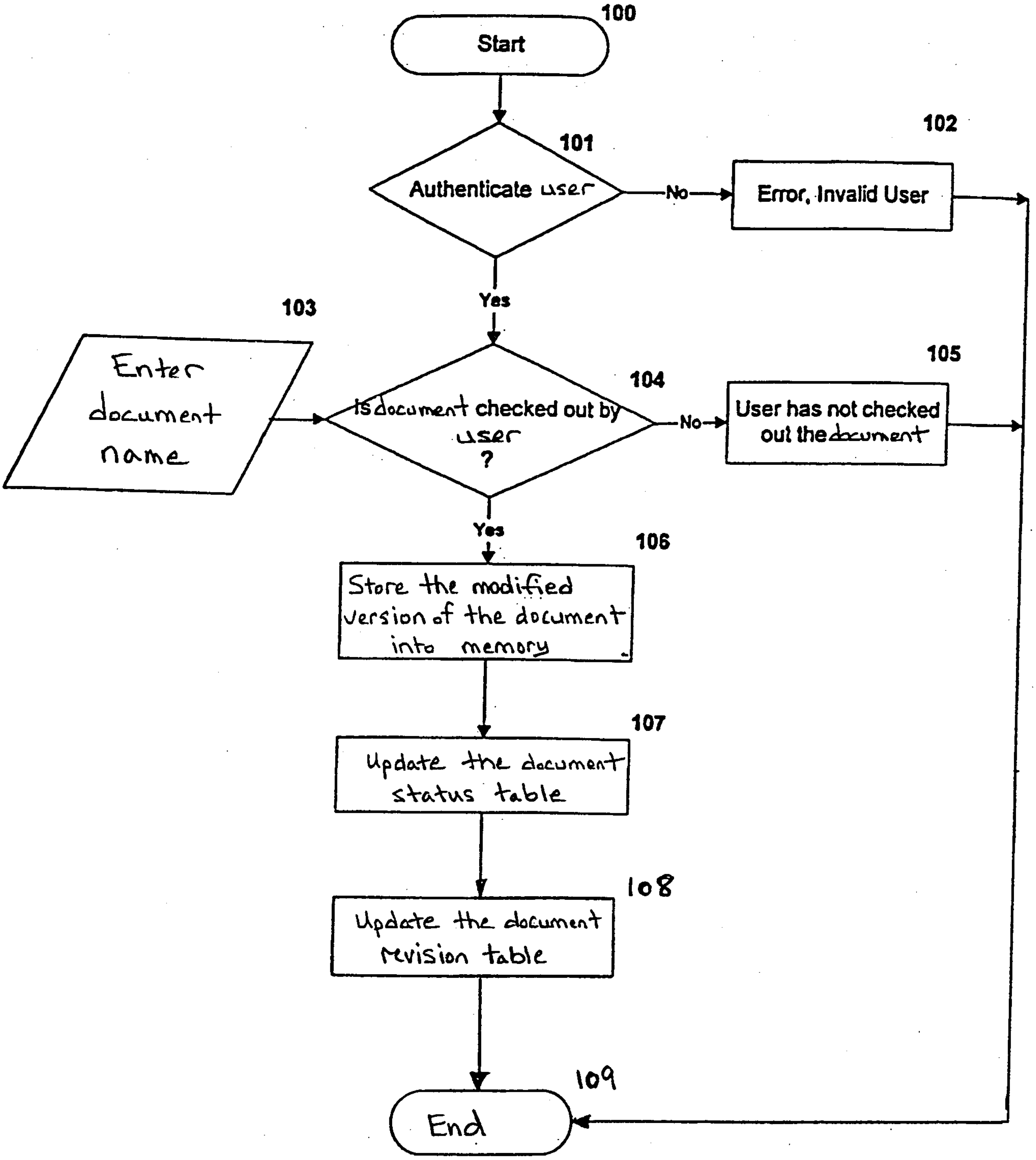
100
Start
101
Authenticate user
No
102
Error, Invalid User
Yes
103
Enter document name
104
Is document checked out by user ?
No
105
User has not checked out the document
Yes
106
Store the modified version of the document into memory
107
Update the document status table
108
Update the document revision table
109
End

FIG. 9

Index	Document Identity	Seller Identity
1	1	1
2	1	2

DOCUMENT STATUS TABLE

FIG. 12

Revision Identity	Revision Number	Document Identity	Seller Identity
1	1	1	1
2	1	1	2

DOCUMENT REVISION TABLE

FIG. 13