

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
12 September 2008 (12.09.2008)

PCT

(10) International Publication Number
WO 2008/109234 A1

- (51) International Patent Classification:
G06K 7/10 (2006.01) *G06K 7/00* (2006.01)
- (21) International Application Number:
PCT/US2008/053841
- (22) International Filing Date:
13 February 2008 (13.02.2008)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
11/683,333 7 March 2007 (07.03.2007) US
- (71) Applicant (for all designated States except US): **WELCH ALLYN, INC.** [US/US]; 4341 State Street Road, Skaneateles Falls, NY 13153-0220 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **WEINER, Herbert, S.** [US/US]; 4647 SE 33rd Avenue, Portland, OR 97202 (US). **PETERSEN, Eric** [US/US]; 19650 SW Madeline Street, Aloha, OR 97007 (US). **FO, Jon, K.** [US/US]; 16060 SW Cormorant Drive, Beaverton, OR 97007 (US).

- (74) Agent: **MIKESELL, Peter, J.**; Hiscock & Barclay, LLP, One Park Place, 300 South State Street, Syracuse, NY 13202 (US).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL,

[Continued on next page]

(54) Title: METHOD FOR PROCESSING DATA FROM BARCODES

WO 2008/109234 A1

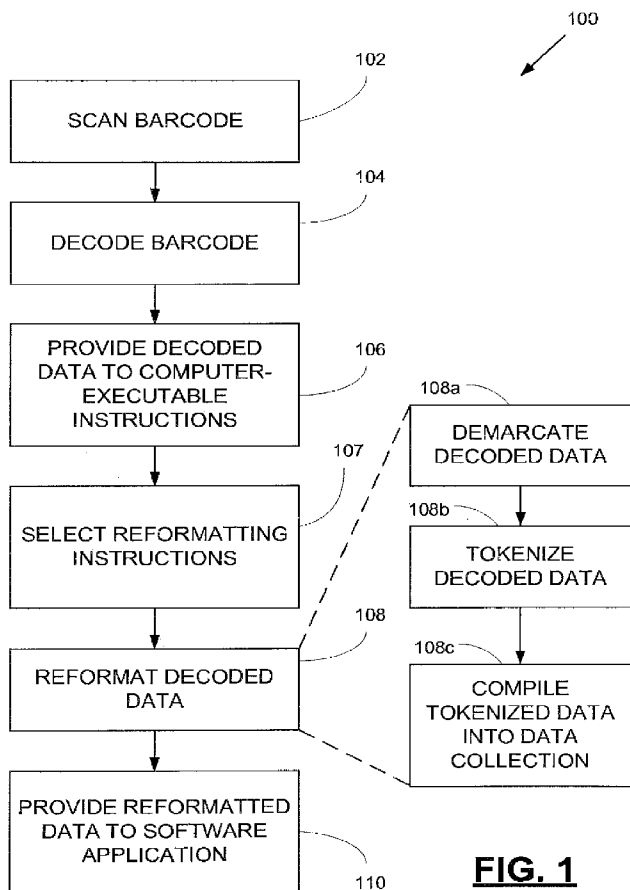


FIG. 1

(57) Abstract: Disclosed in this specification is a method for processing barcode data from one or more barcode scanners. The decoded data is reformatted in accordance with selected reformatting instructions and thereafter supplied to a software application for subsequent use. Preferably, the reformatting instructions can be customized by the end users, thus permitting the end users to adjust their system to receive previously unknown barcode formats.



NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG,
CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

— *before the expiration of the time limit for amending the
claims and to be republished in the event of receipt of
amendments*

Published:

— *with international search report*

METHOD FOR PROCESSING DATA FROM BARCODES

FIELD OF THE INVENTION

[0001] This invention relates, in one embodiment, to a method for processing relevant data from barcodes, regardless of the initial format of such data. Such processed data is preferably provided to a computer, such as the central monitoring station of a medical facility.

BACKGROUND OF THE INVENTION

[0002] Barcodes are widely used in a variety of commercial endeavors. For example, barcodes are commonly used in mailing and shipping businesses to help track packages in transit. Barcodes are also used by retailers such as bookstores, grocery stores, and the like. Generally, barcode usage can be divided into two broad categories wherein: (1) the barcode system is generated to match an industry standard barcode or (2) the organization generates a unique barcode system and barcode for specialized use by the organization.

[0003] Barcode systems are often designed to comply with an industry standard. For example, book retailers design inventory and point of sale systems to handle the industry standard barcodes that publishers imprint on their books. These systems often include proprietary features that are not shared with competitors, for example, to target sales promotions to customers' past purchases; but every retailer's system must be compatible with the industry standard barcodes. Unfortunately, not all industries have established such a standard. In some industries, each organization can simply create unique barcodes.

[0004] A given organization may generate a specialized barcode which is generally only usable by the organization itself. For example, commercial shipping organizations have been known to develop barcode systems for use by their own organization. Since the barcode is only used for internal purposes, the organization may gain a competitive advantage by including proprietary information, such as a route number, on their barcodes and keeping their barcode system proprietary. For example, within the shipping

industry, each individual shipping organization may have its own unique barcode and barcode system which cannot easily be adapted to the barcode system of the competing organizations.

[0005] Context also plays a role in barcode systems. For example, book retailers may scan the barcodes on books when adding them to inventory, and may scan the books again when they are sold. At the time of sale, the retailer may scan barcodes on coupons to determine discounts, and may scan barcodes on preferred customer cards to award bonuses. Clearly, the system must be able to distinguish a book from a coupon from a preferred customer card. When books are added to inventory, the barcode scanning system may look up the wholesale cost of the book, while at the time of sale, the barcode scanning system may look up the retail price. Sophisticated barcode systems use the same barcode in different ways depending upon context, and must be able to distinguish between different types of barcodes, also depending upon context.

[0006] The context may be determined by which system on a network initiates the barcode scan, or by which application program is currently active when the scan is initiated, or by which window in a particular application program is active, or by the location of the cursor within a particular window, or by any combination of these. For example, within a particular window, there may be a field for the entry of products being purchased, and a different field for the entry of coupons or discount codes.

[0007] Unfortunately, difficulties arise when providing products or systems to several organizations which have no established standard. In such situations one can never be certain what format of barcode has already been deployed by the end customer. When the organization in question utilizes products and services that are provided by multiple vendors the situation becomes even more complex. The barcodes required by some systems may be incompatible with the barcodes supported by other systems, necessitating the manual entry of data in situations where the required data is already available on barcodes. The medical care industry is one such industry that suffers from the aforementioned problems.

[0008] In the medical industry centralized patient monitoring stations have begun to appear in hospitals. Such centralized stations receive data from multiple locations and

permit a small number of medical personnel to monitor a large number of patients. Naturally, such stations are heavily computerized. It would be desirable to allow hospitals to enter data into their computer systems without keyboard input. Such an entry system would help to reduce data entry errors. One method for entering data into a computer system without using keyboard input utilizes barcodes.

[0009] While using barcodes to perform keyless entry of data may appear ideal, the medical industry currently suffers from a number of complications that have prevented the implementation of such a system. The primary deficiency is the lack of an accepted barcode standard. For example the hospital may build or acquire a hospital information system, including barcode printing and barcode scanning capabilities, which handles patient admission, discharge, transfer, billing, as well as equipment inventory, personnel assignment, etc. This system likely uses barcodes for patient wristbands, employee badges, equipment inventory tags, laboratory samples, pharmacy labels, etc. The hospital also acquires other equipment, such as patient monitors, infusion pumps, central patient monitoring systems, etc, that typically has not been designed to work with the barcodes used by the hospital information system. Clearly, the hospital is not able to adapt their barcodes to accommodate the requirements of all the other equipment that they acquire. The result is that data such as patient names and ID numbers must, in some cases, be manually entered into the hospital information system (such as central patient monitoring systems), since this equipment is typically not capable of scanning every possible variation of barcode that may be found in various hospitals. Unfortunately, when a wristband is scanned, the central station has no way to determine how to interpret the data. For example, one hospital may use patient wristbands which contain a patient name (last name first) followed by a medical record number, a second hospital may use patient wristbands which contain the medical record number, followed by the patient's date of birth, gender, and patient name (first name first), and a third hospital may encode only the medical record number (without the name) on the barcode. Merely telling the station "Smith,John19420517" is insufficient. A computer views such a string as a mere series of characters. The computer is unable to "see" a name followed by a series of numbers. Additionally, the station cannot determine if the number is a patient identification number, a date of birth, or some other number, nor can the computer differentiate a first

name from a last name. Moreover, some hospitals use barcodes that present data in a format that is different from the barcode format of other hospitals. For example, a patient's name may be presented as "Smith, John," "John Smith," "John B. Smith," "J. B. Smith," or any of the other possible modes of presentation. A similar quandary occurs when considering date formats. Possible date formats include, for example, May 17, 1942; 05-17-1942; 5-17-42; 05-17-42; 17-5-42; 17-May-42, along with many others. When one considers the use of two dimensional barcodes which encode for multiple data elements, the problem of reading multiple barcode formats appears staggering.

[0010] Therefore, a method for processing barcode data is desired wherein the data can be provided to a software application in an acceptable format, regardless of the initial format of the data.

SUMMARY OF THE INVENTION

[0011] The invention comprises, in one form thereof, a system that makes it possible for a central monitoring system of a hospital to easily adapt to the wide range of barcodes deployed in hospitals, so that the barcodes already in use in the hospital can be scanned without modification, and without providing custom software for each hospital.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] The present invention is disclosed with reference to the accompanying drawings, wherein:

Figure 1 is a flow diagram of one process of the invention;

Figure 2, which includes Figures 2A-2E, are examples of decoded data;

Figure 3, which includes Figures 3A and 3B, are additional examples of decoded data;

Figure 4 is a depiction of one process for use with the instant invention and an example of decoded data being subjected to one example of such a process;

Figure 5 is a depiction of a screen with a plurality of data fields to be populated in accordance with the invention;

Figure 6 is an example of one mode of barcode presentation for use with the present invention;

Figure 7 is an example of another mode of barcode presentation;

Figure 8 is an example of yet another mode of barcode presentation and processing;

Figure 9 and Figure 10 are depictions of other modes of barcode presentation and processing, which includes examples of a screen to be populated; and

Figure 11 is an illustration of another screen to be populated in accordance with the instant invention.

[0013] Corresponding reference characters indicate corresponding parts throughout the several views. The examples set out herein illustrate several embodiments of the invention but should not be construed as limiting the scope of the invention in any manner.

DETAILED DESCRIPTION

[0014] Figure 1 is a flow diagram of one method 100 of processing barcode data. Method 100 is initiated by executing step 102, wherein a barcode is scanned with a barcode scanner. Any suitable barcode and barcode scanner may be used. For example, scanners suitable for reading both linear and two dimensional barcodes are contemplated for use with the instant invention. Such barcodes may encode for a single data element or they may include a plurality of data elements. When a barcode is scanned, the barcode provides coded data to the barcode scanner. This coded data is decoded in step 104.

[0015] In step 104 of method 100, the coded data obtained by scanning the barcode is decoded. Typically, the scanner itself is configured to perform step 104, as the precise code being used (i.e. the symbology of the barcode) will vary from barcode to barcode. In other words, the scanner is selected to match the barcode such that the barcode can be

decoded. A variety of suitable scanners are known in the art. Autodiscriminating scanners, which are scanners that can detect and decode multiple symbologies, may also be used in conjunction with the instant invention. Examples of commonly used symbologies include, but are not limited to, Code 16K, Code 39, Code 49, Codabar, Code 128, UPC-E, UPC-A, EAN-8, EAN-13, Reduced Space Symbology, PDE-417, Interleaved 2-of-5, Aztec, and the like. Once the scanner has decoded the coded data, such decoded data is provided to a computer for processing. Such a step is executed in step 106.

[0016] In step 106 of method 100, the decoded data is provided to a computer for processing. Such decoded data is generated by the scanner and presented in a computer-readable format, such as in the American Standard Code for Information Interchange (ASCII) format or ISO Latin 1. The data so provided is preferably stored by the computer in a computer-readable medium, such as in a hard disc, in random-access memory (RAM), an optical storage medium such as a compact disc (CD), or in a similar medium.

[0017] In step 107 of method 100, the computer-executable instructions that are to be used in step 108 are selected from a plurality of computer-executable instructions. Such a selection step may be made based upon a number of criteria. For example, the station where the barcode is located may generate an identifier that causes a specified set of instructions to be selected. Alternatively, the user may specify a set of instructions which is to be applied by interacting with a software application. In another aspect of the invention, the barcode itself includes a data label which serves to identify which set of instructions should be selected. In yet another embodiment, the workflow of the organization determines which set of instructions should be applied. For example, upon the first scan, instructions A are applied and during the second scan instructions B are applied. In this manner a hospital may, for example, sequentially scan a patient's wristband followed by the patient's badge, and both barcodes will be properly processed.

[0018] In step 108 of method 100, the decoded data is reformatted so as to place the data in a standardized data format. The method for reformatting the decoded data is preferably stored as a plurality of computer-executable instructions (e.g. software) stored

on a host computer. Such a host computer is preferably a network server that is operatively connected to multiple barcode scanners. In one embodiment, the network server is the central monitoring station of a medical facility such as a hospital. The decoded data is processed in accordance with the computer-executable instructions and reformatted data is generated. The details of such a reformatting step are discussed in greater detail elsewhere in this specification. Once the data has been reformatted, such data is provided to a software application (step 110) for later use.

[0019] Referring again to Figure 1, the details of reformatting step 108 are depicted in sub-steps 108a, 108b and 108c. In sub-step 108a, the decoded data provided in step 106 is demarcated in accordance with the computer-executable instructions located on the host computer. Figure 2, which includes Figures 2A to 2E, provides examples of such a demarcation step.

[0020] Referring now to Figure 2, the decoded data is demarcated in accordance with the computer-executable instructions. In this specification, the term “demarcate” refers to the act of determining the boundaries surrounding a given “data element.” A “data element” is a series of characters with a logical association such as, for example, a Social Security number, a name, a room number, a date of birth, and the like. A data element may be comprised of other data elements. For example, the data element “Name” may be comprised of the data elements “First Name” and “Last Name.”

[0021] Referring to Figure 2A, a string of numbers is shown. The computer-executable instructions provide the guidance necessary to properly demarcate such a string. For example, in one embodiment, such instructions may demarcate the string every nine characters. In such an embodiment, only a single data element is shown in Figure 2A. In another embodiment, such instructions may demarcate the string between the fourth and fifth character. In such an embodiment, two data elements would be shown in Figure 2A. Some barcodes provide a delimiter which can be used to help demarcate the decoded data. An example of such a delimiter is shown in Figure 2B.

[0022] Referring now to Figure 2B, the decoded data is comprised of a string of characters with carriage returns (<CR>) inter-spliced at certain positions in the string. In one embodiment, the computer-executable instructions interpret such a carriage return as

a delimiter that separates two data elements. In this manner the host computer can determine the location of each of the data elements and properly demarcate each such element. In the embodiment presented in Figure 2B, four such data elements are shown.

[0023] Referring to Figure 2C, an example of an alternate delimiter is given. The embodiment depicted in Figure 2C, the delimiter is one or more blank spaces. For the sake of clarity, such blank spaces are depicted as empty squares. Other delimiters may also be used. Examples of suitable delimiters include commas, blank spaces, tab delimiters, carriage returns, semicolons, a designated string of characters, and the like. In some embodiments, the delimiter is a character (or series of characters) that cannot be found in the character string to be demarcated. So, for example, of spaces may be present in the name field, (*e.g.* Van Dyke) then spaces would preferably not be used as a delimiter.

[0024] In the example illustrated in Figure 2D, no delimiter is present, but the computer-executable instructions dictate that the first data element is given by the first thirteen characters, the second element is given by the next nine characters, and the third element is given by the next six characters. In this manner, the decoded data can be demarcated without using delimiters.

[0025] Referring now to Figure 2E, the decoded data contains data labels that can be used by the computer-executable instructions to help demarcate the data elements. As used in this specification, a "data label" is an indicator that helps a computer assign a logical meaning to a data element. In the embodiment depicted in Figure 2E, the beginning of a data label is denoted by a "<" and the terminus of a data label is indicated by a ">" character. In this manner, the data labels and data elements can be independently demarcated. Data labels may also serve as a type of delimiter, as they can be used to determine the end of one data element and the beginning of the next.

[0026] In one aspect of the invention, the computer-executable instructions are configurable by an end-user after the software has been installed on the central monitoring station. During a typical installation process, a software manufacturer will install the software on a central monitoring station with a specified set of computer-executable instructions. Since, in the instant embodiment, the computer-executable

instructions are configurable by the end-user, it follows that the hospital can modify the formatting instructions without needing to contact the software manufacturer. For example, the formatting instructions may be configurable such that the demarcation criteria is altered. Examples of demarcation criteria are found throughout this specification. For certain applications, such a configuration increases the usefulness of the invention.

[0027] In one embodiment, the computer-executable instructions are configurable through an end-user-interface. Such an end-user-interface permits the end-user, who may not necessarily be a computer programmer, to configure the computer-executable instructions. Similar user-interfaces are well known in the art, and often find application in macro writing.

[0028] Referring again to Figure 1, once the decoded data has been properly demarcated (sub-step 108a) then sub-step 108b is executed, wherein the demarcated data is tokenized. Reference may be had to Figure 3.

[0029] Figure 3, which includes Figures 3A and 3B, is an example of the tokenizing of data elements. Referring to Figure 3A, the data has been demarcated into three data elements using spaces as the delimiter prior to being tokenized. The term “tokenize” refers to the assignment of a logical meaning to a given data element and giving a computer variable the value of the data element. For example, the data element “123456789” may be tokenized to be given the meaning of a Social Security number, the computer variable “SSN” is assigned the value of “123456789.” Alternatively, such an element could be tokenized to be given the meaning of a patient identification number. In accordance with the computer-executable instructions, the first such element is tokenized to be the patient’s name. Such a data element maybe further demarcated and tokenized such that the element is comprised of two sub-elements, delimited by a comma. The first such sub-element is the last name, and the second such sub-element is the first name. The second data element is tokenized to be the patient’s Social Security number. The third such data element is tokenized to be the room number. In the embodiment depicted in Figure 3A, such tokenizing is determined based upon the order of the demarcated data. In Figure 3B, a data label assists such tokenizing.

[0030] In Figure 3B, two data elements and two data labels are shown. In the embodiment depicted in Figure 3B the data element is tokenized in accordance with the computer-executable instructions such that its assigned value is dictated by the data label which precedes such element. For example, the data label “<NAME>” precedes the data element “Smith,John” and therefore such element is tokenized to be the patient’s name. Likewise, the data label “<SSN>” is tokenized to be the patient’s Social Security number. The patient’s name may, in turn, be tokenized into sub-elements as previously described.

[0031] Referring again to Figure 1, once the demarcated data has been properly tokenized (sub-step 108b), the tokenized data elements are compiled into reformatted data collection. Such a compilation occurs in sub-step 108c. A more detailed depiction of sub-step 108c is given in Figure 4.

[0032] As shown in Figure 4, sub-step 108c is comprised of sub-steps 402 and 404. In sub-step 402, selected elements from the list of tokenized data elements are reformatted. For example, in the embodiment depicted in Figure 4, data collection 406 contains three data elements, the first of which has been tokenized to indicate it is the patient’s name. In data collection 406, the patient’s name is in the format “last name, first name.” As previously discussed, such a data element has also been tokenized into sub-elements to indicate the first and last name of the patient. During reformatting sub-step 402, the two sub-elements have their relative order altered, and thus produce data collection 408. In data collection 408, the patient’s name has been reformatted to be in the format “first name last name.” Additionally, the trailing spaces have been truncated. Additionally, the patient’s Social Security number (shown in data collection 406) is initially a simple a string of numbers. In data collection 408, the Social Security number has been annotated to include hyphens at character positions four and seven. In a similar manner, the patient’s room number has been annotated to include additional characters such as the words “building” and “room.” In addition to truncation and annotation, the data elements may be edited in other ways. For example, two data elements may be concatenated into a single element. Alternatively or additionally, a data element may have selected characters deleted. In another embodiment, not shown, the entire data element may be deleted. Once sub-step 402 has been completed, data collection 410 is produced by executing sub-step 404.

[0033] Referring again to Figure 4, once the individual data elements have been reformatted in sub-step 402, sub-step 404 is executed wherein the reformatted data elements are compiled into reformatted data collection 410. In the embodiment depicted in Figure 4, several changes occurred during sub-step 402. For example, the order of the data elements was altered during compilation such that the Social Security number is now the first data element, and the patient's reformatted name is now the second element. Additionally, a delimiter has been added after each data element. In the embodiment depicted, the delimiter is a "TAB" indicator. In other embodiments, not shown, multiple delimiters are inserted after certain data elements. Lastly, in sub-step 404, an End-of-File (EOF) marker has been added to indicate the end of the present data collection 410.

[0034] The following examples are provided to help clarify certain aspects of the present invention. Such examples are merely explanations of certain embodiments. Such explanations should not be interpreted to limit to scope of the invention to such embodiments.

EXAMPLE 1

[0035] Example 1, shown in Figure 6, will now be described with reference to Figure 5, wherein a plurality of barcodes are provided each of which encode for a single data element. In the instant embodiment, computer-executable instructions dictate that the decoded data should be annotated with a <TAB> marker at the end of each data collection. In the present example, each of the four barcodes shown in Figure 6 contain a single data element.

[0036] Referring now to Figure 5, screen 500 is the screen of a computer monitor on a central monitoring station at a hospital. Screen 500 has a plurality of fields for displaying data. Such fields include Social Security field 502, name field 504, room field 506, and insurance field 508. The software application which populates the aforementioned fields is configured such that the user can press the tab key to move between the fields in the aforementioned order. The user places the cursor at first cursor position 510 such that Social Security field 502 is selected. Thereafter, the user scans

barcode 600 (see Figure 6) which is the encoded Social Security number for the patient at issue.

[0037] Referring now to Figure 6, barcode 600 is the encoded Social Security number for the patient at issue. The user scans barcode 600 and the barcode scanner generates decoded data 602. Such decoded data is provided to the central monitoring station. The central station recognizes that the cursor is presently located in the Social Security field and recalls the computer-executable instructions that apply the appropriate reformatting commands. In the present embodiment, the reformatted data 604 is identical to decoded data 602 except in that a <TAB> delimiter is added at the end of the data collection. The formatted data is then provided to the software application.

[0038] Referring again to Figure 5, when the software application receives reformatted data 604, the Social Security field 502 becomes populated with the value "123456789" just as if the user had typed the data. When the software application receives the <TAB> delimiter, the software application advances the cursor to the next data field - name field 504. Name field 504 is now the currently selected field.

[0039] Name field 504 is then populated in an analogous manner by scanning barcode 606. The scanner produces decoded data 608. The central station recalls the appropriate computer-executable instruction and generates reformatted data 610. In the present embodiment, such instructions are the same for all fields - a <TAB> delimiter is added.

[0040] When the software application receives reformatted data 610, the name field 504 (see Figure 5) becomes populated with the value "Smith, John" just as if the user had typed the data. When the software application receives the <TAB> delimiter, the software application advances the cursor to the next data field - room field 506. Room field 506 is now the currently selected field.

[0041] Similarly, room field 506 and insurance field 508 are populated by scanning barcodes 612 and 614. In the present embodiment, the insurance field 508 will contain the value "56." Such a value is a unique primary key that is used by the central station to cross-reference data corresponding to the desired insurance carrier.

EXAMPLE 2

[0042] Example 2, shown in Figure 7, will now be described wherein a single barcode 700 is provided which encodes for a plurality of data elements. In the instant embodiment, the computer-executable instructions dictate that the current delimiter, a carriage return, should be replaced with a <TAB> marker at the end of each data element.

[0043] Referring again to Figure 5, the cursor is placed at first cursor position 510 such that the Social Security number field 502 is selected. In the present embodiment, when the decoded data 702 is provided to the central station, the station recalls the appropriate computer-executable instruction that corresponds to screen 500. In accordance with the instructions, the carriage returns are replaced with <TAB> delimiters and data collection 704 is generated. The final carriage return is not replaced, but is simply deleted. When the data collection 704 is provided to the software application fields 502, 504, 506, and 508 each become appropriately populated. Since the delimiter is a <TAB>, and since the software application interprets the <TAB> as an instruction to move to the next field, the user can populate all four fields by scanning a single barcode 700.

EXAMPLE 3

[0044] Example 3, shown in Figure 8, is substantially similar to Example 2, except in that the barcode 800 is provided by a different manufacturer. The instant software application (see Figure 5) anticipates receiving decoded data in a certain order. Specifically, the software application expects to receive the patient's Social Security number, followed by the patient's name. The manufacturer who generated barcode 800 provides the decoded data in an order which is not initially compatible with the present software application illustrated in Figure 5. Reformatting is used to rectify this deficiency.

[0045] In the instant embodiment, barcode 800 is scanned and decoded data 802 is produced. The relative order of the name and social security data elements is altered

during compilation of the data collection 804. Additionally, the computer-executable instructions dictate that the current delimiter, a carriage return, should be deleted and a <TAB> marker should be annotated at the end of each data element. In this manner the data collection 804 is placed in a format that is suitable to populate screen 500. The data collection 804 is then provided to the software application as previously discussed.

EXAMPLE 4

[0046] Example 4, shown in Figure 9, is similar to Example 3, except in that the barcode 900 is provided by yet another manufacturer. Unlike Example 3, the software application which receives the formatting data contains supplemental hospital field 908 which is not presented in the decoded data. The computer-executable instructions demarcates and tokenizes the name field into a first name element and a last name element suitable to populate fields 904 and 906. The Social Security number is annotated by including hyphens. Reformatting is again used to address these deficiencies.

[0047] Referring now to Figure 9, in the embodiment depicted therein, barcode 900 is decoded to generate decoded data 912. The central station detects the instant software application (see Figure 5) and the barcode scanner terminal which provided the decoded data 912 and recalls the appropriate computer-executable instructions. When such instructions are applied to decoded data 912, reformatted data collection 914 is generated. In accordance with the instructions, the fourth data element (corresponding to hospital field 908) has been assigned a null value, as the barcode 900 does not contain the data necessary to populate such a field. When reformatted data collection 914 is provided to the software application, fields 902, 904, 906, 908 and 910 are appropriately populated as previously described. Hospital field 908 contains the null value. Supplemental data, such as hospital field 908, may be entered manually.

EXAMPLE 5

[0048] Example 5, shown in Figure 10, is similar to Example 4, except in that the barcode 1000 is provided by yet another manufacturer. Unlike Example 4, the barcode

1000 contains additional data (a street address) which is not used by the software application. Reformatting is again used to rectify this deficiency by omitting the unused element during the reformatting step.

[0049] Referring again to Figure 10, decoded data 1004 contains the patient's name, address, and social security number. The software application detects which barcode scanning terminal is providing the decoded data 1004 and recalls the appropriate computer-executable instructions. During the reformatting step, the decoded data 1004 is reformatted to generate data collection 1006 which is in a format suitable to populate the fields shown in Figure 10. Since the screen does not contain an address field, during compilation of the data collection 1006, the address data is omitted. Like the previous examples, the order of the data elements was altered and the empty data elements, such as hospital field 908, was populated with null values.

EXAMPLE 6

[0050] Example 6, not shown, is similar to Example 5, except in that the empty fields are later populated by scanning additional barcodes.

[0051] In the present Example, the process is initiated in a manner identical to that described for Example 5. Once the Social Security number, first name and last name fields are populated, the user places the cursor within hospital field 908 (see Figure 10). The user then scans a separate "hospital" barcode which populates hospital field 908 in a manner similar to that described in Example 1.

EXAMPLE 7

[0052] Example 7, not shown, is similar to Example 6 in that an additional barcode is scanned to populate the hospital field. However, in the present example, the decoded data which is provided by the scanner is comprised of a data label "<Hospital_ID>" which is used by the central station to identify the data. The central station then populates hospital field 908 (see Figure 10) with the appropriate data, even if the cursor was not placed within hospital field 908 prior to scanning the barcode.

EXAMPLE 8

[0053] Example 8, not shown, generates a data collection which is stored as a data file. Such a data file is later queried by one or more software applications for the data elements contained therein. A new data file may be created or the data collection may be appended to or inserted into an existing file. The data collection may add data elements to the data file, or replace existing data.

[0054] In Example 8, barcode data is scanned as described in the other Examples. Example 8 differs from Examples 1-7 in that the data is not displayed, but is imported directly to a software application which organizes and saves the reformatted data collection in a data file. In one embodiment, the data elements are stored in the data file in a predetermined order, such that a second software application can query the data file and retrieve the desired data element(s). In another embodiment, the data elements are annotated with data labels in the data file to facilitate such retrieval. In yet another embodiment, both data labels and a predetermined order are used to facilitate retrieval of data. In one embodiment, an End of File (EOF) marker is annotated to the last data element in the data collection to denote the terminus of the data file. The data file may be a plurality of related files or tables, such as a relational database.

EXAMPLE 9

[0055] In Example 9, shown in Figure 11, a single central networked computer station is operatively connected to a plurality of client terminals with barcode scanners. The barcode scanners are configured to read barcodes provided by different manufacturers and provide the decoded data to the central station over the network. The central station detects which formatting instruction should be applied, recalls the appropriate computer-executable instruction, and generates a reformatted data collection. In one preferred embodiment, the computer-executable instructions are loaded on the networked computer. The reformatted data collection is then provided to a software

application. Such a software application may be, for example, a discrete program, a subroutine, a module of code, and the like.

[0056] Referring again to Figure 11, a software application generates screen 1100 which is sub-divided into patient screen 1102, hospital screen 1104 and medication screen 1106. Each of such screens include one of the radio buttons 1108, 1110, and 1112. A user selects one of the radio buttons and thereby makes the selected sub-screen the currently selected screen. Based upon which screen is currently selected, the central station recalls a specified computer-executable instruction from a plurality of computer-executable instructions. In other embodiments, not shown, such a selection is made by the software application itself based upon the detection of which barcode scanning terminal is providing the data collection. For example, all data coming from the pharmacy barcode scanning terminal may be automatically treated as pertaining to medication screen 1006, and the appropriate reformatting instructions are applied. In one such embodiment, the decoded data provided by the scanner includes an indicator which is used by the terminal to properly identify the correct set of computer-executable instructions.

[0057] Referring again to Figure 11, screen 1100 is populated by selecting radio button 1108 which makes patient screen 1102 the currently selected screen. Accordingly, the computer-executable instructions for such patient data are recalled. When the central station receives decoded data from a barcode scanner, the proper reformatting commands are applied and fields 1114 and 1116 are populated. For example, the first data element received (the Social Security Number) may have hyphens included at positions four and seven.

[0058] After patient screen 1102 is populated, hospital screen 1104 is selected by activating radio button 1110. Hospital screen 1104 becomes the currently selected screen. Accordingly, the computer-executable instructions for such screen are recalled. In the instant Example, such instructions differ from the instructions used to format screen 1102.

[0059] In the present example, medication screen 1106 contains data concerning the medications the patient is currently taking. Such data is entered into the central

monitoring station by scanning the barcodes on the medicine containers. In the present example, the barcodes on the medicine containers encode for Universal Product Codes (UPC) using the Code 39 symbology. At the hospital of the instant example, the only barcodes which use Code 39 symbology are the medicine containers. The barcode scanner terminal at the hospital's pharmacy, which originally scanned the container, provides an appropriate indicator to the central station. The central monitoring station detects that the decoded data was originally encoded in Code 39 symbology. Accordingly, radio button 1112 is automatically selected by the central monitoring station, and the fields in screen 1106 are populated after applying the appropriate computer-executable instruction to format the data.

[0060] While the invention has been described with reference to preferred embodiments, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted for elements thereof to adapt to particular situations without departing from the scope of the invention. Therefore, it is intended that the invention not be limited to the particular embodiments disclosed as the best mode contemplated for carrying out this invention, but that the invention will include all embodiments falling within the scope and spirit of the appended claims.

What is claimed is:

1. A computer system for processing barcode data comprising the steps of:
 - a. scanning a barcode with a barcode scanner, thereby obtaining coded data,
 - b. decoding the coded data, thereby producing decoded data,
 - c. providing the decoded data to a computer,
 - d. reformatting the decoded data in accordance with a set of computer-executable instructions which contains reformatting instructions by
 - i. demarcating the decoded data into one or more data elements in accordance with the selected computer-executable instructions, thus producing one or more demarcated data elements,
 - ii. compiling at least one of the data elements in accordance with the selected computer-executable instructions to produce a reformatted data collection,
 - e. providing the reformatted data collection to a software application.
2. The computer system as recited in claim 1, wherein the step of reformatting includes the step of tokenizing at least one of the demarcated data elements in accordance with the selected computer-executable instructions, thereby producing one or more tokenized data elements.
3. The computer system as recited in claim 1, wherein the computer-executable instructions are selected from a plurality of computer-executable instructions located on the computer.
4. The computer system as recited in claim 1, wherein the barcode is a linear barcode with a plurality of data elements.
5. The computer system as recited in claim 1, wherein the barcode is a two dimensional barcode with a plurality of data elements.
6. The computer system as recited in claim 1, wherein the barcode is a linear barcode with a single data element.

7. The computer system as recited in claim 1, wherein the barcode is a two dimensional barcode with a single data element.
8. The computer system as recited in claim 1, wherein supplemental data, which is not contained in the barcode, is manually entered into the software application.
9. The computer system as recited in claim 1, further comprising the step of configuring at least one computer-executable instruction from the plurality of computer-executable instructions such that the reformatting of the decoded data is altered by an end-user.
10. A computer-readable medium having software instructions for performing a method comprising the steps of:
 - a. receiving decoded data from a barcode scanner,
 - b. reformatting the decoded data in accordance with a set of computer-executable instructions which contains reformatting instructions by
 - i. demarcating the decoded data into one or more data elements in accordance with the selected computer-executable instructions, thus producing one or more demarcated data elements,
 - ii. tokenizing at least one of the demarcated data elements in accordance with the selected computer-executable instructions, thereby producing tokenized data elements,
 - iii. compiling at least one of the tokenized data elements in accordance with the selected computer-executable instructions to produce a reformatted data collection,
 - c. providing the reformatted data collection to a software application.
11. A computer readable medium as recited in claim 10, wherein the set of computer-executable instructions is selected from a plurality of computer-executable instructions.

12. The computer readable medium as recited in claim 11, wherein at least one of the plurality of computer-executable instructions is configurable by an end-user after installation of the software instructions.
13. The computer readable medium as recited in claim 12, wherein at least one of the computer-executable instructions is configurable such that the end-user controls the reformatting of the decoded data.
14. The computer readable medium as recited in claim 13, wherein at least one of the computer-executable instructions is configurable such that the end-user controls the reformatting of the decoded data by specifying demarcation criteria.
15. The computer readable medium as recited in claim 13, wherein the computer-executable instructions are configurable such that the end-user controls the reformatting of the decoded data by specifying at least one delimiter.
16. The computer readable medium as recited in claim 13, wherein the computer-executable instructions are configurable such that the end-user controls the reformatting of the decoded data by specifying at least one data element which is to be deleted from the reformatted data collection.
17. The computer readable medium as recited in claim 10, wherein the reformatting step further comprises the step of demarcating at least one demarcated data element to produce demarcated sub-elements.
18. The computer readable medium as recited in claim 17, wherein the reformatting step further comprises the step of altering the relative order of the demarcated sub-elements.
19. The computer readable medium as recited in claim 10, wherein the reformatting step further comprises the step of altering the relative order of the demarcated elements.

20. The computer readable medium as recited in claim 10, wherein the step of selecting the computer-executable program includes the step of determining which barcode scanning terminal provided the decoded data.
21. The computer readable medium as recited in claim 10, wherein the step of selecting the computer-executable program includes the step of determining the identity of the software application.
22. The computer readable medium as recited in claim 10, wherein the software application includes a current cursor position and the step of selecting the computer-executable program is based upon the current cursor position.
23. The computer-readable medium as recited in claim 10, wherein the step of reformatting further comprises the step of annotating the data collection.
24. The computer-readable medium as recited in claim 23, wherein the step of annotating includes adding a data label to the data collection.
25. The computer-readable medium as recited in claim 23, wherein the step of annotating includes adding a delimiter between each of the data elements.
26. The computer-readable medium as recited in claim 25, wherein the delimiter is selected from the group consisting of a comma, a tab, a semicolon, a carriage return, and combinations thereof.
27. The computer-readable medium as recited in claim 10, wherein the decoded data is comprised of at least three data elements.
28. The computer-readable medium as recited in claim 10, wherein the barcode scanner scanned a barcode and generated coded data with a symbology, wherein the computer-executable instruction is selected based on the symbology of the barcode.
29. The computer-readable medium as recited in claim 10, wherein the step of demarcating the decoded data includes the step of demarcating a data label associated with the demarcated data elements.

30. The computer-readable medium as recited in claim 29, wherein the computer-executable instruction is selected based on the data label.
31. A computer-readable medium having computer-executable instructions for performing a method comprising the steps of:
- a. receiving decoded data from a barcode scanner, wherein the decoded data is comprised of a plurality of data elements,
 - b. selecting a computer-executable instruction from a plurality of computer-executable instructions, wherein the plurality of computer-executable instructions each contain reformatting instructions,
 - c. reformatting the decoded data in accordance with the selected computer-executable instructions by
 - i. demarcating the decoded data into a plurality of data elements in accordance with the selected computer-executable instructions, thus producing a plurality of demarcated data elements,
 - ii. tokenizing at least one of the demarcated data elements in accordance with the selected computer-executable instructions, thereby producing tokenized data elements,
 - iii. compiling at least one of the tokenized data elements in accordance with the selected computer-executable instructions to produce a reformatted data collection, wherein the reformatted data includes a delimiter separating the data elements, and the data elements are in an order specified in the selected computer-executable instructions,
 - d. providing the reformatted data collection to a software application, wherein the software application has data formatting requirements, and the selected computer-executable instructions were selected to satisfy the formatting requirements.
32. The computer-readable medium as recited in claim 31, wherein the step of demarcating the decoded data includes searching the decoded data for a specified data label.

33. The computer-readable medium as recited in claim 31, wherein the decoded data is comprised of a source indicator wherein the computer-executable instruction is selected based on the indicator.

1/10

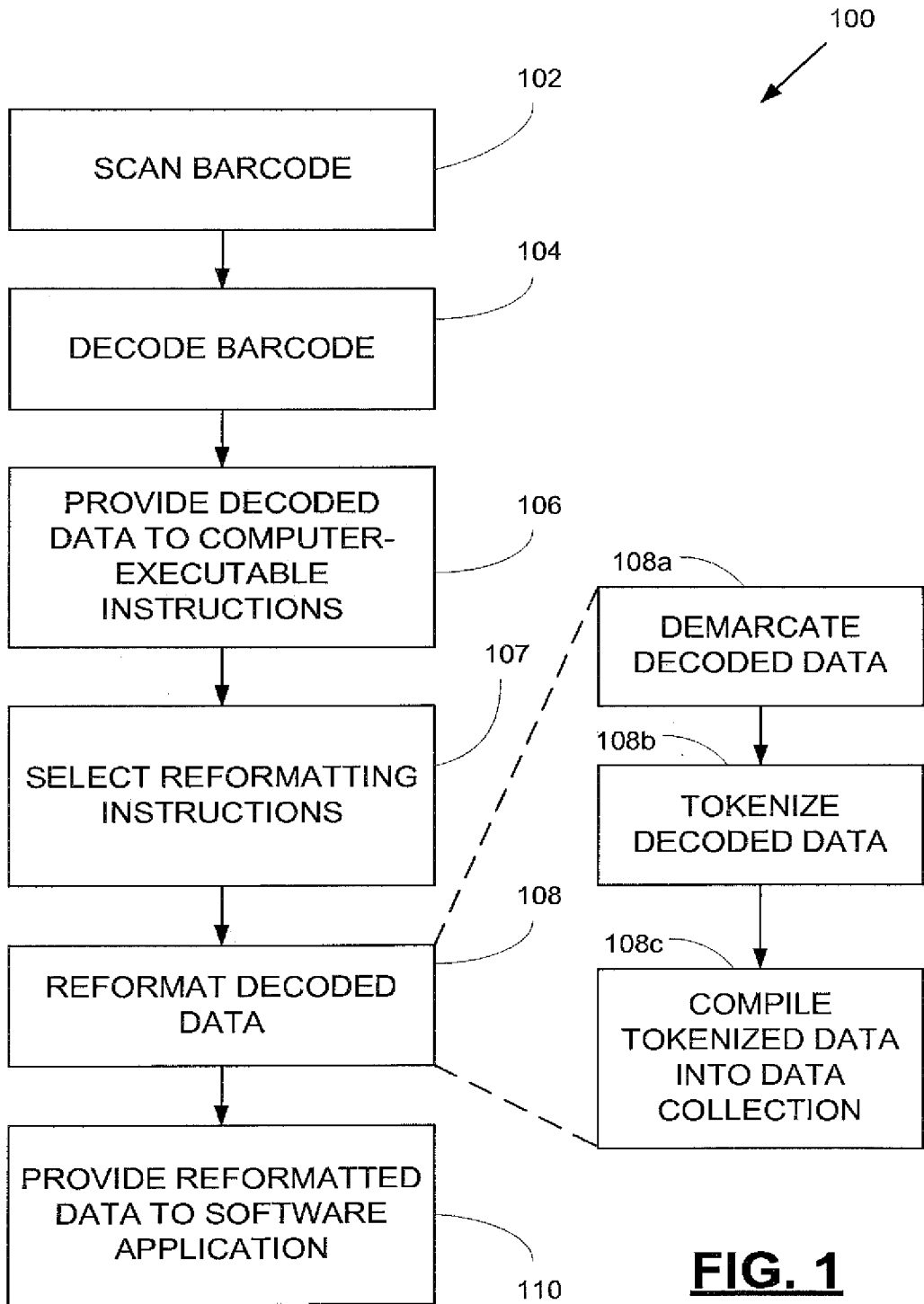


FIG. 1

2/10

FIG. 2A

123456789

FIG. 2

FIG. 2B

Smith,John<CR>123456789<CR>9876-A<CR>4226<CR>

FIG. 2C

Smith,John□123456789□9876-A

FIG. 2D

Smith,John□□□1234567899876-A

FIG. 2E

<Name>Smith,John<SSN>123456789<Room>9876-A

FIG. 3A

Smith,John□123456789□9876-A
SSN

FIG. 3

FIG. 3B

<Name>Smith,John<SSN>123456789
Label Element Label Element

3/10

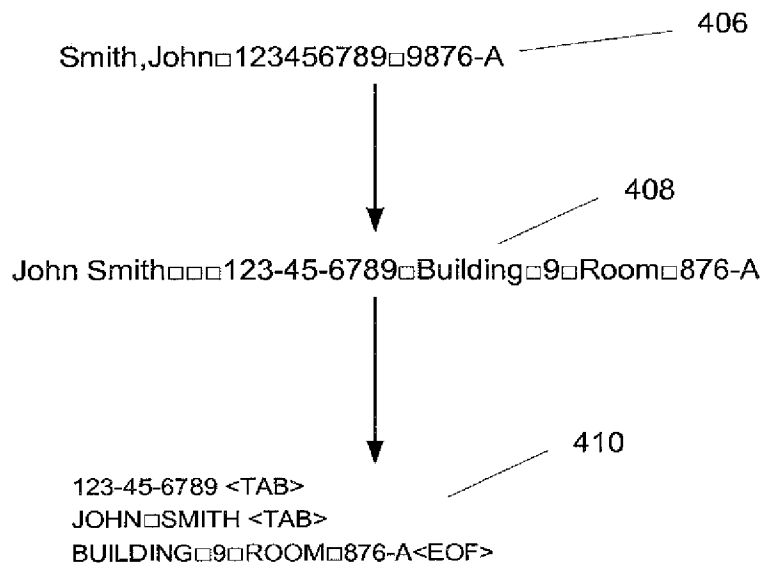
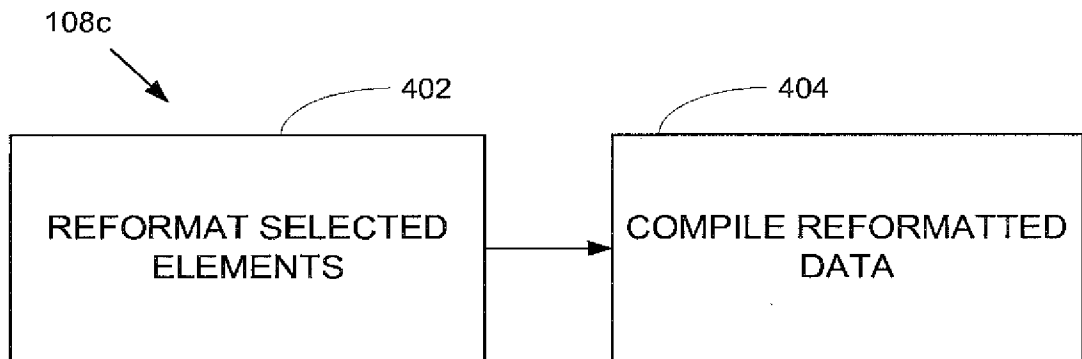


FIG. 4

4/10

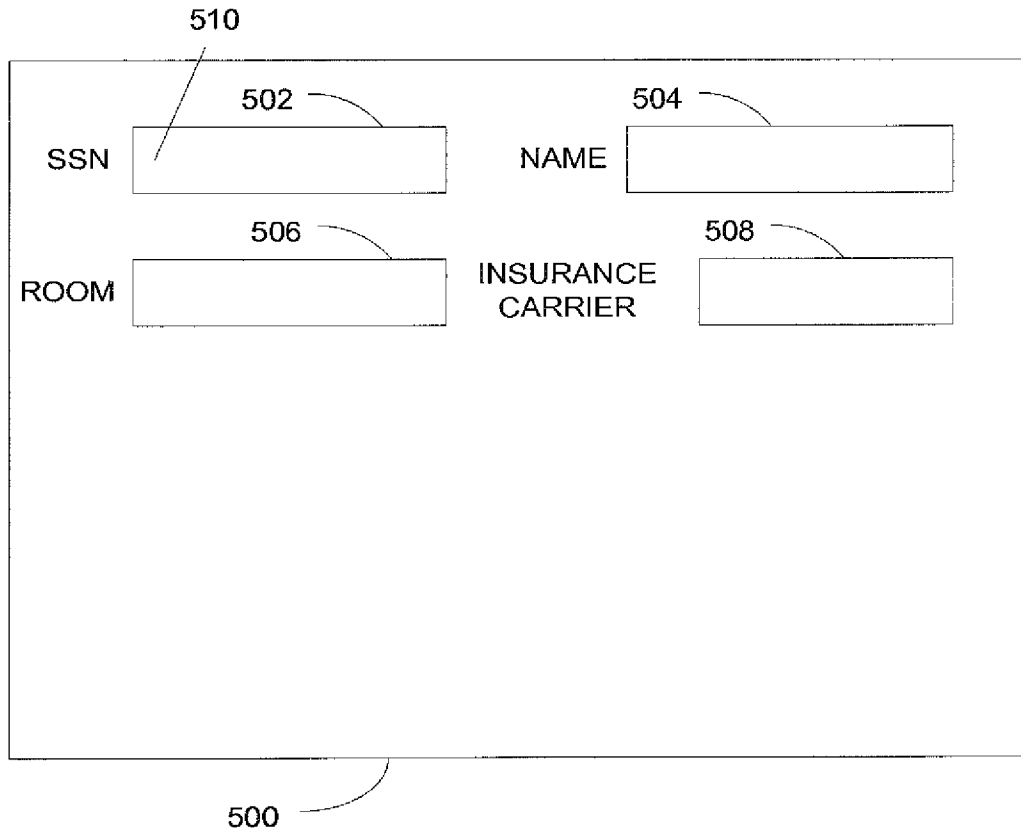


FIG. 5

5/10

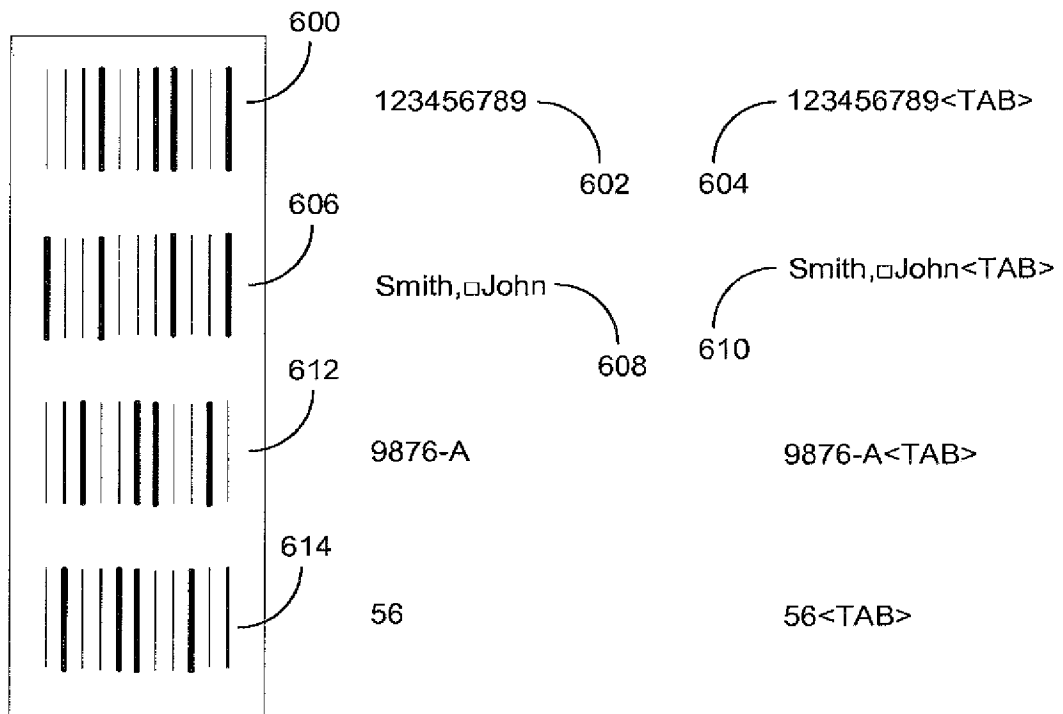


FIG. 6

6/10

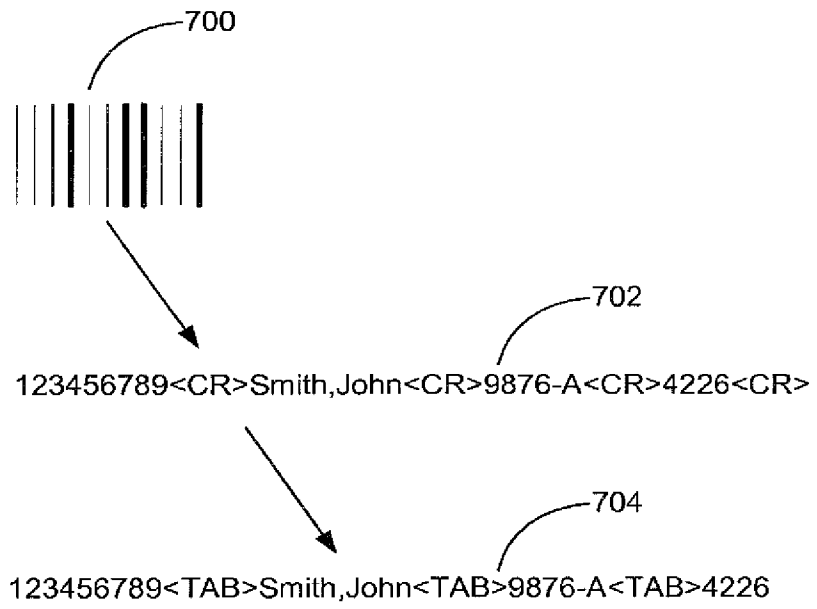


FIG. 7

7/10



FIG. 8

8/10

SSN	<input type="text"/>		
First Name	<input type="text"/>	Last Name	<input type="text"/>
Hospital	<input type="text"/>	Room	<input type="text"/>

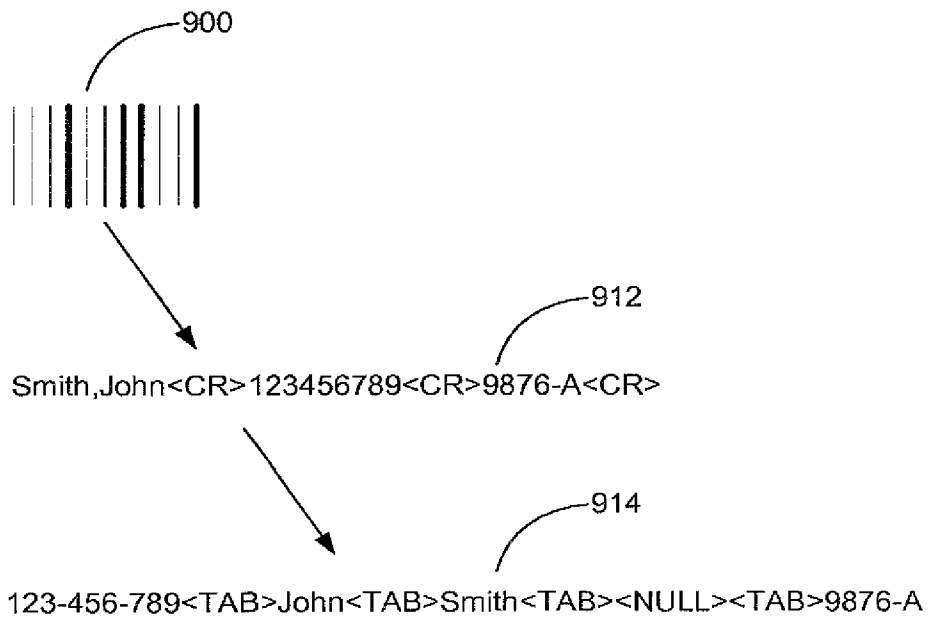


FIG. 9

9/10

SSN	<input type="text"/>		
First Name	<input type="text"/>	Last Name	<input type="text"/>
Hospital	<input type="text"/>	Room	<input type="text"/>

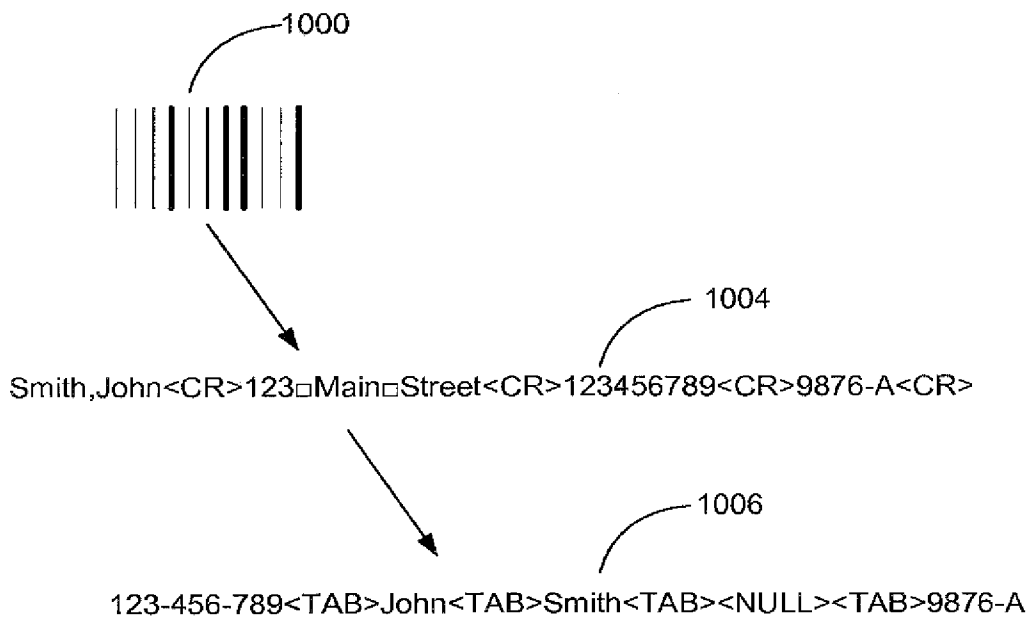


FIG. 10

10/10

1100

1108

1114

1116

SSN

NAME

1102

1110

Hospital

Room

1104

1112

Medication



1106

Detailed description: The figure shows a form with three horizontal sections. The top section contains two input fields: 'SSN' with reference numeral 1114 and 'NAME' with reference numeral 1116. The middle section contains two input fields: 'Hospital' and 'Room'. The bottom section contains one wide input field labeled 'Medication'. Reference numeral 1108 points to a small circle in the top-left corner of the first section. Reference numeral 1110 points to a small circle in the top-left corner of the second section. Reference numeral 1112 points to a small circle in the top-left corner of the third section. Reference numeral 1102 points to the right edge of the top section. Reference numeral 1104 points to the right edge of the middle section. Reference numeral 1106 points to the right edge of the bottom section.

FIG. 11

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2008/053841

A. CLASSIFICATION OF SUBJECT MATTER		
<i>G06K 7/10(2006.01)i, G06K 7/00(2006.01)i</i>		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) IPC 8: G06K		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Korean Utility models and applications for Utility models since 1975 Japanese Utility models and applications for Utility models since 1975		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) eKIPASS(KIPO Internal) "barcode", "scan", "decode", "reformat", "compile"		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6517002 B1 (JOHN T. PIATEK) 11 February 2003 See claims 1-6, column 4, line 56 - column 5, line 59, figure 2	1-33
A	US 7097099 B2 (PAUL HEPWORTH et al) 29 August 2006 See claims 1-3, column 5, line 52 - column 6, line 46, figure 1	1-33
A	US 5536928 A (JEREMY SEIGEL) 16 July 1996 See claims 1-4, column 2, line 66 - column 3, line 21, figure 2	1-33
A	US 6540144 B1 (DONALD T. HUDRICK et al) 1 April 2003 See claim 1, figure 2	1-33
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 15 JULY 2008 (15.07.2008)		Date of mailing of the international search report 16 JULY 2008 (16.07.2008)
Name and mailing address of the ISA/KR  Korean Intellectual Property Office Government Complex-Daejeon, 139 Seonsa-ro, Seo-gu, Daejeon 302-701, Republic of Korea Facsimile No. 82-42-472-7140		Authorized officer JEON, CHANG IK Telephone No. 82-42-481-8261 

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2008/053841

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6517002 B1	11.02.2003	CA 2264242 A1	26.09.1999
US 7097099 B2	29.08.2006	US 2004-0158604 A1 US 2006-0138236 A1 WO 2005-076873 A2 WO 2005-076873 A3	12.08.2004 29.06.2006 25.08.2005 23.03.2006
US 5536928 A	16.07.1996	US 5514859 A	07.05.1996
US 6540144 B1	01.04.2003	None	