Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 3028906 C 2022/01/11

(11)(21) **3 028 906**

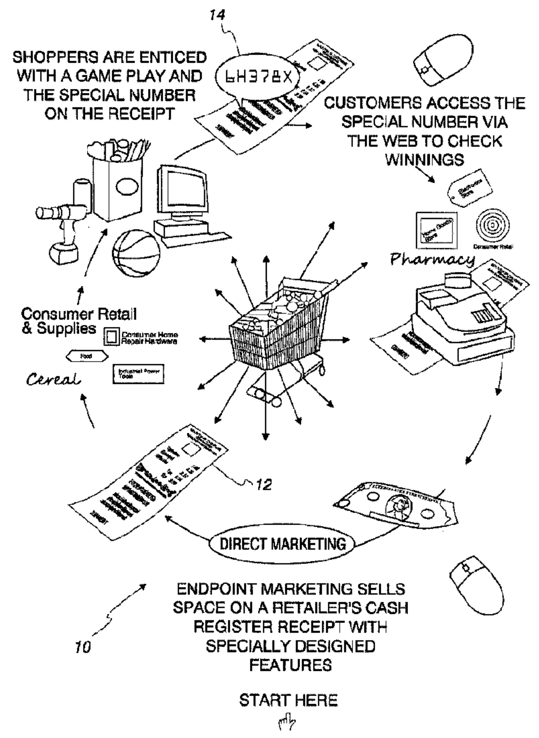
(12) **BREVET CANADIEN**
**CANADIAN PATENT**

(13) **C**

(22) Date de dépôt/Filing Date: 2019/01/04
(41) Mise à la disp. pub./Open to Public Insp.: 2019/07/10
(45) Date de délivrance/Issue Date: 2022/01/11
(30) Priorité/Priority: 2018/01/10 (US15/866,996)

(51) Cl.Int./Int.Cl. *G06Q 30/02* (2012.01)
(72) Inventeur/Inventor:
DORIS, ERROL S., SR., US
(73) Propriétaire/Owner:
DORIS, ERROL S., SR., US
(74) Agent: ANDREWS ROBICHAUD

(54) Titre : METHODE ET APPAREIL DE PROMOTION DES VENTES ET D'AUGMENTATION DE LA RECONNAISSANCE DU NOM DE MARQUE
(54) Title: METHOD AND APPARATUS FOR PROMOTING SALES AND INCREASING BRAND NAME RECOGNITION


14
SHOPPERS ARE ENTICED WITH A GAME PLAY AND THE SPECIAL NUMBER ON THE RECEIPT
LH378X
CUSTOMERS ACCESS THE SPECIAL NUMBER VIA THE WEB TO CHECK WINNINGS
Pharmacy
Consumer Retail & Supplies
Cereal
12
DIRECT MARKETING
ENDPOINT MARKETING SELLS SPACE ON A RETAILER'S CASH REGISTER RECEIPT WITH SPECIALLY DESIGNED FEATURES
10
START HERE

(57) Abrégé/Abstract:
A method for promoting sales and increasing brand recognition is described. The method includes selecting a brand name based on customer purchases enumerated on a transaction receipt generated on basis of transactions performed by a particular customer. The brand name is then obfuscated. The customer is provided an application to be used in conjunction with the obfuscated brand name to try and complete the missing letters in the brand name. The customer is rewarded with an incentive for uncovering the brand name and in order to submit the solution the customer is obligated to provide demographic information.

# METHOD AND APPARATUS FOR PROMOTING SALES AND INCREASING BRAND NAME RECOGNITION

## BACKGROUND OF THE INVENTION

1. Field of the Invention.

[01] This invention relates to a method for promoting brands at point of sale scenarios, and more specifically, the invention uses cash register receipts and coupons as promotion vehicles for various well-known brand names.

2. Background of the Invention.

[02] Brick and mortar retailers process in access of 45 billion cash register transactions each year. There are more than 310,000 cash registers at these retailer locations in the United States.

[03] Millions of stock keeping units (SKUs) vie for shelf space in stores all over the world; all in an effort to expand brand recognition and therefore sales. Retail chain stores average about 45,000 SKU's per location. However, only so much space is available in typical brick and mortar retail outlets. The average Big Box retailer stocks an inventory of between 90,000 and 120,000 items in warehouses. These items are sold exclusively by internet.

[04] A need exists in the art for a vehicle to expand brand exposure without concomitant increase in allocation of shelf space. The vehicle should be visible to every purchaser visiting brick and mortar establishments. The vehicle should also engage the purchaser and ideally leverage that purchaser's penchant for certain products.

## SUMMARY OF INVENTION

[05] An object of the invention is to provide cost effective brand exposure in brick and mortar establishments that may overcome many of the disadvantages of the prior art.

[06] Another object of the present invention to provide a method for pre-selling revenue futures to retailers. A feature of the invention is the use of cash register

receipts to leverage a purchaser's preference for certain products. An advantage of the invention is that retailers share in advertising proceeds that are generated when the cash register receipt is used as an advertising vehicle.

[07] Still another object of the present invention is increasing advertising through purchaser incentives and participation. A feature of the invention is using point of sale advertising vehicles identifying buyer preferences to induce customers to participate in a contest. An advantage of the invention is that point of sale vendor support is provided in exchange for creating an attractive audience for brand managers of those retail outlets.

[08] Yet another object of the present invention is to increase exposure of customer-preferred advertised brands by utilizing a fast response customer experience via a smart phone and other intelligent devices. To facilitate this use, the customer will utilize an "app" linking to a web portal displaying a game play incentive program that uses the brand names as features of the game play.

[09] Briefly, the invention provides a method for promoting sales and increasing brand recognition, the method comprising selecting brand names of products based on customer purchases enumerated on a customer receipt, wherein a specific alpha-numeric sequence appears on that receipt; obfuscating one of said brand names on the customer receipt so as to induce a customer to attempt to guess the one of said brand name; and determining if the customer guessed correctly. Each winning is determined by the customer logging on to a web portal and entering the special code displayed on the cash register receipt.

[10] Also provided is a brand-name promotional system, the system comprising a means (e.g. a scanner) for recording customer purchases at a retail outlet; means for categorizing product classes based on the customer purchases; game piece generator for printing indicia onto a customer receipt; wherein the indicia is an obfuscated brand name found in the product classes; and means for determining if a customer correctly guesses the brand name from the indicia. The system also provides a method to retrieve the customer's winning remotely, for example via a web portal.

## BRIEF DESCRIPTION OF DRAWING

[11] The invention together with the above and other objects and advantages will be best understood from the following detailed description of the preferred embodiment of the invention shown in the accompanying drawings, wherein:

[12] Fig. 1 depicts a flow chart showing a summary of one embodiment of the system, showing player, controller, advertiser, and retailer functions;

[13] Fig. 2A depicts an embodiment of static paper receipt from a retailer per one embodiment of the invention;

[14] Fig. 2B depicts a sample of a digital receipt from a retailer per one embodiment of the invention;

[15] Fig. 3 depicts a sample screen pursuant to one embodiment of the invention;

[16] Fig. 4 depicts a flow chart of operations of one aspect of one embodiment of the invention;

[17] Fig. 5 depicts a flow chart of operations of one aspect of one embodiment of the invention;

[18] Fig. 6 depicts a flow chart of operations of one aspect of one embodiment of the invention;

[19] Fig. 7 depicts example vouchers used in conjunction with one embodiment of the invention;

[20] Fig. 8 depicts a flow chart of operations of one aspect of one embodiment of the invention;

[21] Fig. 9 depicts a sample interface for one embodiment of the invention;

[22] Fig. 10 depicts a flow chart of operations of one aspect of one embodiment of the invention;

[23] Fig. 11 depicts stages of gameplay of one embodiment of the invention;

[24] Fig. 12 depicts a flow chart of operations of one aspect of one embodiment of the invention;

[25] Fig. 13 depicts a flow chart of operations of one aspect of one embodiment of the invention;

[26] Fig. 14 depicts an example interface, per one embodiment of the invention;

[27] Fig. 15 depicts further example interfaces, per one embodiment of the invention;

[28] Fig. 16 depicts an example interface, per one embodiment of the invention;

[29] Fig. 17 depicts an example interface, per one embodiment of the invention;

[30] Fig. 18 depicts an example interface, per one embodiment of the invention;

[31] Fig. 19 depicts an example interface, per one embodiment of the invention;

[32] Fig. 20 depicts an example interface, per one embodiment of the invention;

[33] Fig. 21 depicts a flow chart of operations of one aspect of one embodiment of the invention;

[34] Fig. 22 depicts a flow chart of operations of one aspect of one embodiment of the invention; and

[35] Fig. 23 depicts a flow chart of operations of one aspect of one embodiment of the invention;

## DETAILED DESCRIPTION OF THE INVENTION

[36] The foregoing summary, as well as the following detailed description of certain embodiments of the present invention, will be better understood when read in conjunction with the appended drawings.

[37] To the extent that the figures illustrate diagrams of the functional blocks of various embodiments, the functional blocks are not necessarily indicative of the division between hardware circuitry. Thus, for example, one or more of the functional blocks (e.g. processors or memories) may be implemented in a single piece of hardware (e.g. a general purpose signal processor or a block of random access memory, hard disk or the like). Similarly, the programs may be stand-alone programs, may be incorporated

as subroutines in an operating system, may be functions in an installed software package, and the like. It should be understood that the various embodiments are not limited to the arrangements and instrumentality shown in the drawings.

[38] As used herein, an element or step recited in the singular and preceded with the word "a" or "an" should be understood as not excluding plural said elements or steps, unless such exclusion is explicitly stated. Furthermore, references to "one embodiment" of the present invention are not intended to be interpreted as excluding the existence of additional embodiments that also incorporate the recited features. Moreover, unless explicitly stated to the contrary, embodiments "comprising" or "having" an element or a plurality of elements having a particular property may include additional such elements not having that property.

[39] The invented method controls promotion of various products. The method provides a means for revenue building for retailers from their existing vendor bases. A cornerstone of the method is to induce cash receipt holders (i.e., purchasers of goods) to utilize the cash receipt as a game piece. An exemplary inducement is to associate the cash receipt with a popular board game. In an embodiment of the method, the cash receipt displays the board game name in its familiar logo type.

[40] Retail outlets are suitable venues for practicing the method. Exemplary retailer types include home improvement stores, supermarkets, grocery stores, pharmacies, general stores, electronic stores, computer stores, variety goods stores, department stores, and club membership stores. In 2012, these retailers executed more than 45 billion cash transactions at over 47,000 locations, each with an average of nine point-of-sale (POS) cash registers or terminals.

[41] FIG. 1 is a schematic depiction of the invented method, the method designated as numeral 10. A first step in the method is to produce a cash register receipt with printed indicia 12. In one embodiment, the indicia will be specific for the purchaser's type of goods bought. In another embodiment, the indicia will have no bearing on any of the current goods purchased and will not be based on the class of goods just purchased. However, in another embodiment indicia is specific to type of store where the receipt was printed and in other embodiments, the "class" of goods will be reflected in the indicia. For example, in use of this embodiment in a supermarket

setting, there are three classes: (a) food/beverage (b) health & beauty (c) household. The home improvement store uses three exemplary classes as well (a) tools (b) hardware (c) appliances. Typically with a promotional coupon program, no one provides incentive at the point of sale with immediate savings, the coupon provider incentivizes the customer to make a return store visit that day, or another day, or online, to purchase the items promoted.

[42] Aside from the standard printed data on the receipt, such as items bought, price, tax, return policies, etc., the receipt will include two unique additions which transforms it into a game piece. First, the receipt will have an obfuscated version of a brand name the purchaser has seen before. In an embodiment of the invention, the brand name is selected based on the database established for that particular purchaser. In another embodiment of the invention, both the retailer and the advertiser is responsible for populating the database.

[43] Second, the receipt will also have a number string, word string, or an alpha-numeric string 14. Ostensibly, this second indicia will allow purchasers access to a website to check their winnings when they have attempted guessing the brand name from the obfuscated version of the brand name on their cash receipt. However, the second indicia will, just as importantly, originate from a plurality of unique identifier strings associated with a brand name guessing game, for example the Brandoodles® board game. The board games come in both virtual and hardcopy form. Virtual ones allow the board game to be structured toward the individual purchaser's buying preferences, and the database of that purchaser's repeated purchases and visitations to the website.

[44] Generally, the game is developed in different media forms. For example, as an embodiment in a board games such as Brandoodles®, which is owned by BELCOM CORP. of Chicago, IL, and comprises a brand name guessing game.

[45] Another embodiment comprises a television game show. In this embodiment, the audience utilizes a smart phone or intelligent device to participate onsite or remotely from the show venue.

[46] Yet another embodiment is an online version of a well-known board game, such as Brandoodles®. A multi-faceted web portal hosts the player access portal.

Retrieval of winnings occurs by means of coupons or codes based on historical data maintained by advertisers and retailers. That data is retrieved and applied to the on-line interaction via a standard data link. The database of advertised products is further determined and modified as part of the progression of the game.

[47] FIGS. 2A and B are schematic diagrams of exemplary cash register receipt/game pieces 12. A top portion 16 contains typical purchase details and return policy details. A bottom portion 18 of the cash register receipt 12 contains the aforementioned two unique additions which transforms the receipt into the game piece. The first game piece contains the obfuscated brand name 13. In the example depicted in FIG. 2A, the brand name is MAKITA CORDLESS DRILL, and therefore an appropriate genre for the type of store (e.g. a food store) which generates the particular cash register receipt or another type of transaction receipt or other form of proof of purchase. FIG. 2B contains the obfuscated brand name 13 MAKITA CORDLESS DRILL which would be appropriate for any type of shopper, in any type of store establishment that generates such a cash register receipt or other form of proof of purchase such as a digital order confirmation. The functions responsible for the receipt/game pieces 12 are depicted as the system 39, described in detail below.

[48] The second edition is the aforementioned number string, word string or alpha-numeric string 14. The specific alpha-numeric string depicted (6H378X) is for illustrative purposes only and not intended to limit the invention to this symbol sequence. In one embodiment, the system employs commercially available software to generate millions of randomly generated strings. The association of the alpha-numeric string and Brandoodles® is based on (a) the assignment of select alphanumeric codes that is assigned a specific value by an administrator account, such as administrator@brandoodles.com in one embodiment and (b) the pre-sale of a bucket of codes being sold by the administrator@brandoodles.com to the advertiser and/or the retailer for the purpose of participation._The codes can be generated using a string generation software, such as an online string generator found in several locations such as at the sweepjudge.com string generator, the source code of which is attached hereto as Appendix B.

[49] In operation, a game administrator (e.g., a licensee of the invented process) charges advertisers (e.g. brand managers) a fee to advertise their brand name on a cash register as a receipt or digital proof of purchase. The retail outlet is charged a few hundreds of a U.S. cent as a "click charge" to access this revenue generator. The retailers agree to use the face of their proof of purchase system as a portal to the process. The retailers and brand managers make product selections, and determine the number of stores involved and the duration of involvement.

[50] The game administrator manages the pre-sells (e.g., the aforementioned click charges) and generates the random numbers 14 appearing on the cash register receipts. A method for transferring the random numbers is for the administrator to load them into the retailer- or brand manager's headquarter-database.

[51] The game administrator then converts the pre-sold number 14 to game play. Winning numbers are preselected. In an embodiment of the invention, a brand manager pre-selects groups of the winning numbers according to an algorithm that generates the numbers according to the odds of winning. In this case, everyone becomes a winner. For example, if 1000 products are preselected to be displayed, the odds shall be designated according to a discounting process, given the distribution sample as follows:

| 30 wins | 50 wins | 70 wins | 90 wins | 110 wins | 130 wins | 150 wins | 170 wins | 200 wins |
|---|---|---|---|---|---|---|---|---|
| 90% | 80% | 70% | 60% | 50% | 40% | 30% | 20% | 10% |

[52] Therefore, if a Retailer sold a product for $20, 30 wins at a 90% discount would be $18 off the price at which the buyer would only pay $2. The Buyer would receive a retrieval code with a value of $18 at redemption. Likewise, 200 wins at a 10% discount, the Buyer would receive a discount of $2 off the price at which the buyer would pay $18 at redemption.

[53] Hence, if the Seller is an Automotive Dealer selling a big ticket item such as a Chevrolet Cruze with a list price of $17,130, the administrator has the option to change the algorithm according to the expected giveaway value, as follows:

[54]

| 3 wins | 5 wins | 7 wins | 15 wins | 30 wins | 40 wins | 200 wins | 300 wins | 400 wins |
|---|---|---|---|---|---|---|---|---|
| 90% | 80% | 70% | 60% | 50% | 40% | 30% | 20% | 10% |

[55] In this case 3 wins at a 90% discount would save $15,417 and 400 wins at a 10% discount would save $1,730 which would be redeemed with a code at the time of purchase.

[56] When reconciled by the winner shopper, the winner is linked to the administrator's website which in turn notifies the retailer and/or brand manager of the existence of the winner. Auditing procedures are established to confirm winner versus loser rates. In one embodiment, the exclusive game administrator is administrator@brandoodles.com. In another embodiment of the invention, both the administrator's website and the retailer's website (and perhaps the brand manager's website) are notified of the existence of a winner.

[57] The invented method and system for promoting brand names is based on incentive. Retail outlet customers are induced to play the game embodied on their cash receipt/game piece when they see a well-known logo of a brand name guessing game.

[58] Retailers are induced to participate in the invented method and system inasmuch as some of the advertising proceeds are funneled back to the retail outlets.

Example

[59] Users of the invention can access commercially available copies of the Brandoodles® game system, or similar system at several game stores or outlets via a web search or physical search of "big box" retail locations. For example, the latest electronic form of an embodiment of the invention can be found online. The source code of the splash screen (FIG. 3) of this web based embodiment is included below as Appendix A. The screen features a plurality of play toggles 31-36 the toggles designed to actuate the game. A panel 35 of letters is provided to allow the user to try and guess the letters associated with the brand name depicted.

[60] The Player's board is an interactive platform that will generate hash algorithm applications of alpha characters based on a random generator with a fixed or

variable value. The fixed value may be 26 – representing the number of letters in the English alphabet. The fixed value may be 30 – representing the number of letters in the Spanish alphabet. The fixed value may be variable with different languages with additional subsets of characters. Hence, If the brand in question being used is designated as having four words, a brand example may be: TROPICANA PREMIUM ORANGE JUICE. This brand possesses 4 words with 30 characters, the game piece generator may produce a possible $(26)^4$ power or a possible randomization of 456,976 combinations of words revealed on each surface that the game is being played, such as the Actual Final Result shown:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| T | R | O | P | I | C | A | N | A |
| P | R | E | M | I | U | M | | |
| O | R | A | N | G | E | | | |
| J | U | I | C | E | | | | |

[61] The Random Generator would show each result differently as shown on each surface such as:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| T | | O | | I | C | | N | A |
| | R | | M | | U | M | | |
| O | R | | N | | E | | | |
| | | I | C | E | | | | |

[62] And again in every instance as shown here:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| T | | O | P | | | A | | A |
| P | | E | | I | U | M | | |
| O | | A | | G | E | | | |
| J | U | | C | | | | | |

[63] The uniqueness of this generation is that no two game sets will appear alike on any surface of the smart phone, computer or other intelligent device, hence, the randomization will also apply to the output on every cash register receipt as different.

*Overview of Operation*

[64] Turning to Fig. 4, depicted therein is an overview of the details of the operation of one embodiment 10 of the system.

[65] The game begins at stage 40. In one embodiment the game enters stage 40 when a buyer completes a transaction at a point of sale terminal and is issued a printed receipt with a game piece showing letters and a code to enter. Example of such receipts are shown herein. In another embodiment, the game enters the start 40 phase when an end user visits a website which includes an electronic display of the game 10.

[66] In the embodiment shown in Fig. 4, in order for the game to proceed from the start state 40 to the begin gameplay state 44, an administrator must elect the category and the number of words to be used in the game 10. In the embodiment shown in Fig. 4, the administrator choices are performed independently of the user playing the game, and this step will occur before a game begins at state 44. The administrator of the game performing the administrative duties of state 42 can make changes to the game concurrently with a number of players, as the information about a particular promotion is stored within the system independently of any players' actions.

[67] Once at least one category and words are chosen in the administrative step 42, the gameplay can begin at the gameplay state 44.

[68] For embodiments where a player was not provided with a preprinted receipt, the system 10 performs a random game piece generator at randomizing state 46. The algorithm at the randomizing state 46 generates unique individual random choices from amongst letters of the alphabet that will appear on the four individual rows of data. Again, in an embodiment where the player has received a pre-printed game piece on a receipt, at state 46 does not generate a new game piece. The randomization state 46 references a database of brands 48 to generate the random puzzle board.

[69] Once the puzzle board is either generated, or an existing one is used from a receipt, a timer 50 starts. The system 10 then proceeds to allow the user to select letters at state 52. As shown in the remaining figures, the user is provided with a letter board and is asked to fill in missing letters from the game piece. The letter choosing state 52 is limited by the timer 50.

[70] During the letter choosing state 52, the system 10 also selects a number of letters for the specific game board in play at help state 54. In the help state 54 the system selects unique random choices from among the letters of the alphabet appearing on a single individual row to choose up to seven hints of letters that have not yet appeared in the row of data. The help state 54 provides both the random hints 56 randomly selected at the help state 54 and the actual instructions for the player at the instructions state 58.

[71] During the letter choosing state 52 the player will fill in the letters of the game piece either by choosing letters directly or by using the assistance of the help state 54. If all the letters are revealed by the end user or the end user wishes to solve the puzzle early, the system 10 moves to the solve state 60. In the solve state 60 the timer 50 is stopped and the end user is asked to fill any all the remaining letters. The proposed solution is then submitted to the system 10.

[72] The system 10 then decides whether the proposed solution is considered a win 66 or a loss 74. In one embodiment the system 10 determines that a system is a win 66 if all the letters match exactly the puzzle. In another embodiment, the embodiment 66 will accept phonetic equivalents of the puzzle solution and therefore will consider a proposed solution 60 a win 66 even if some of the letters are not exactly correct. For example, in some instances a particular puzzle may comprise a product name which has alternative spellings, such as one in French and English or English and Spanish.

[73] If a proposed solution is determined to be a winning solution at state 66, the puzzle is revealed at the revelation step 64. The system then proceeds to the prize calculation step 62. The details of the calculation step 62, and other algorithms described below, are presented in the algorithm in Appendix C. At the prize calculation step 62, the system 10 calculates the player's odds based on the value of the prize and

number of players and awards a quantity in each percentile value. Once the prize calculation step 62 assigns a prize to the winning game, the game proceeds to calculate the particular player's odds 68. Finally, the game 10 proceeds to the prize award state 70.

[74] As shown in Fig. 4 while the player is solving the puzzle 60 the timer 50 continues to run. If the game enters the timer ended state 72, the player is given one more chance of solving the puzzle at state 60.

[75] If upon submitting a solution at state 60 the system 10 determines that the proposed solution is incorrect, the system moves to the incorrect state 74. The player is offered an opportunity to play the game again at state 76. If the player agrees to play the game again at state 79, the system 10 returns to the begin state 40. If the player chooses to not play again at state 78, the game ends 80.

[76] In one embodiment the choice to play again at the play again state 76 is offered to the player only if the player has not repeated plays an excessive number of times in a given time period. In another embodiment, the play again state 76 will only provide the option to repeat gameplay if the system administrator at the administration step 42 has entered additional puzzles for the end user.

[77] In one embodiment, the end user who started game play using a printed receipt will only be able to repeat games using the products reflected on the player's receipt. In this case, the interaction between the database of brands 48 and the selection algorithm 46 also interacts with the receipt information gathered in the begin state 40.

[78] As can be appreciated from Fig. 4, the system 10 provides for a method of generating and completion of puzzles, including puzzles that relate to a database of brand names 48.

## *Transaction Receipt Numbers*

[79] Turning to the flowchart of Fig. 5 depicted therein is a schematic for the process of selecting random numbers which act as transaction receipt numbers. In one embodiment, the transaction receipts are cash register receipts. In another

embodiment, the transaction receipts take the form of electronic or other form of purchase confirmation.

[80] As explained herein, the system 10 includes a game piece attached to a receipt, such as the sample receipt shown in Fig. 2. The receipt includes a redemption code 14. The steps of assigning the code and related tasks, per one embodiment 90, are shown in Fig. 5.

[81] The process 90 begins at a start state 92. The start state 92 comprises a website administration portal 94 in one embodiment. As part of the website administrator portal 94 the system administrator can select a product category and product names 96. The selection of product information 96 results in updates to the database of retailers 98 and the database of brands 100. Once the databases 98, 100 are updated, the system administrator can select the brand which will be shown the to end user as part of the promotion as part of step 102. The administrator can also chose the number of products 104 that will be subject to the promotion.

[82] Once the brands 102 and the products 104 are chosen, the system 90 assigns the random alphanumeric codes at step 106. The randomization step 106 processes and assigns random alphanumeric codes that are transmitted to the cash register receipt.

[83] Once the randomization step 106 is completed the system 90 proceeds to identify a retailer 108 which will use a particular set of puzzles for products. In one embodiment, as part of selecting the retailer 108, the system will also include metadata regarding the promotion, including the start time and end time as well as the start date and end date for each potential promotion.

[84] Once the target retailers are identified 108, the promotions and all related metadata are sent to the retailer's database in step 110.

[85] Concurrently with adding brand and product choices, the administrators 94 can observe when random codes are redeemed online using interface 116. Random codes are assigned 112 from the retailer's database 110 by the retailer's IT system. The assignments 112 connect a particular redemption code to the receipt and it is acquired by the customer 114. The administrators 94 can see how the codes are redeemed 116 as part of the game starting process 40 shown in Fig. 4.

[86] While the various data stores such as the databases 110, 98, 100 are depicted as separate objects in the flow chart in Fig. 5, in one embodiment the data stores are located on a single computer which comprises the embodiment of the system 90. Further while the data stores are shown as distinct databases in the Fig. 5, in various embodiments the databases are data entries in a single data store, either an organized database or other form of data storage, both organized and not. In one embodiment the data stores are implemented using a distributed data store such as a blockchain record store.

*Game Play Opening Step*

[87] Turning to Fig. 6 depicted therein is a flow chart of the game play opening steps. The game play begins with the game software 140 being invoked on one or more devices 142.

[88] To begin playing, the end user must register or log in to the game using the login step 144. To register, the end user must provide identifying information such as first and last name as well the email address, a phone number, date of birth, and gender, in the depicted embodiment. Once the end user provides the identifying information, the end user can create a username for the system at step 146. After creating the username, the end user can choose preferences 148 and either redeem a cash register or point of sale wining code 148 or redeem an automatic winning voucher 150.

[89] If after establishing a username the end user successfully logs into the system in order to play 154, the end user can redeem one or more the vouchers 150, 152 or begin playing the game 156. The end user can select one or more game play styles 158, each of which is described herein. As discussed in this application, the end user may play multiple games and can, in some instances, repeatedly play the promotional games. However, if the player chooses not to play the game, the system 140 will end the game and log the end user out at stage 159.

[90] Turning to Fig. 7, depicted therein are several winning vouchers as described in steps 150, 152. The example voucher 160 includes a redemption code 162 and a UPC code 164. The UPC code can be used in a point of sale terminal. For

awards that do not involve a point of sale terminal, other redemption instructions 166 will be used, as is the case with the instructions for a promotion of a vehicle's price shown in Fig. 7.

[91] Fig. 8 in turn depicts the details for the registration and login process 170 which is invoked from the application 140. The login process 170 allows the end user to log in using a unique identification number 172. In order to finalize the account the end user must also provide a first name 174 and a last name 176 and an email address 178. The email address is confirmed. The end user also provides a cell phone number 180.

[92] When the end user logs into the system and confirms their email address the end user may reset their password or proceed to log in to the system and request login information.

[93] As part of the account set process and log in process the end user must also review the terms and the disclaimer 182 and agree 184 to the process of the using the application 140.

[94] As the game 140 requires the date of birth for its players, the log in process requests that the players confirm the date of birth and provide the date of birth 186 when necessary. Further, as the promotions are often gender specific, the gender information is also collected from the end user.

[95] In one embodiment, the step of requesting the date of birth 186 comprises asking the end user to select a date range representing end user ages, instead of requesting the specific date of birth. In another embodiment, the step of requesting the date of birth 186 comprises requesting the specific birthday but not year. In this case the system is able to provide the end user promotions relating to the end users' birthday, but without requiring the exact birth year for every end user. In one embodiment, the end user is provided with a range of incentives to provide the birth year and birthday information, such as a promise of additional promotions if birthday and year information is given.

[96] The conclusion of the process is the creation of a system identifier 188 which allows the end user to interact with the system.

[97] Fig. 9 in turn depicts several interfaces for the game 140 to login and register in the system. The login in steps 170 are depicted in the interface 190. The

interface accepts from the user the name 192, email address 194, and phone number 196 using a similar look and feel as the game pieces shown to the end user during actual game play.

[98] Once the fields of the interface 190 are completed, the system proceeds to the disclaimer of rights 182 and other steps as shown in Figs. 8 and 9. In conjunction with the age usage restrictions, the system also display specific notices about cell phone restrictions and regulatory notices such as limits of promotional games in Canada. The end user is provided an opportunity to agree to the various terms and restrictions 184. If the end user agrees to continue, the system will establish an identity for the user 198. The end user will then be provided with an interface to confirm the birthdate 200. Finally, the end user will see their assigned system identifier 202. In the embodiment shown in Fig. 9, the identifier comprises the initials, an encoded birthdate, a flag for the gender, and an incremental counter, as well as a country identifier.

[99] In another embodiment the identifier 202 includes a random number. In yet another embodiment the identifier is simply a random number not already assigned an existing user.

[100] Fig. 10 depicts another login or registration process. In this process the application 140 login 170 accepts as input an identification number which is verified 210 by the application 140. The identification number which is verified originates from a third party, such as a point of sale purchase, a loyalty rewards program, or others. The verified 210 identifier represents a category of rewards 212 such as a discount code or an instant prize. The expiration date of the identification number is confirmed. The end user may apply the identified reward as a gift 214 to a third party or to redeem it 216 for that particular user.

[101] Regardless of whether it is to the end user's account or to a third party account as a gift, the reward points are applied to a balance 218. The balance 218 is redeemed online 220 or in-store 222. In either case, the final transaction ends in checkout of the purchase including the awarded claim.

[102] If the beneficiary of the verified identifier 210 is not currently a member of the system 140 the registration information may be received 224 by the system 140 and

the system identifier 226 may be created as described above before any redemption occurs.

[103] Turning to Fig. 11, depicted therein are several example interfaces of the system 140. The interfaces shown depict the game play steps A-F up to and including providing the end user the option to redeem the code online 220 or in-store 222.

*Dashboards*

[104] Fig. 12 depicts the operational details of an interface or dashboards 230. One dashboard 232 contains information about the controller of the game including making pre-selections for promotions. The controller dashboard 232 interacts with the data network 249 to communicate with one or more data stores containing information about the game. The dashboard 232 includes information about each product category 234, the number of products 234 and a brand alphabet 236. The system also generates the random brand alphabet 238, the random percentage values 240. The points of sale 242 are established. The demographic data 244 is liked to particular promotions and rewards program 246 data is also linked to promotions. Finally, participants are consigned reward points 248.

[105] The dashboard 230 also includes a means for the end user to login or register 250 using a tablet, laptop or handheld computing device 252. The end user can begin a game 254. The game can involve a cash register or point of sale receipt puzzle 260 or where the end user begins the puzzle from an online game source 256. The end user must fill in the missing letters during the available time period set by a timer 258. The winning brand is awarded 262 with cross merchandizing premium added 264, if earned. The cross merchandising premium is unlocked by accessing a special character 268 in the puzzle board. The system involves award points 266 and an add on premium brand award 270, in one embodiment.

[106] A second set flowcharts for user dashboards are depicted in Fig. 13. The dashboards again require a login 280. The entities that can log in to the system using this dashboard 280 include registered retailers 282 and registered advertisers 284. Once the retailers or advertisers are logged in the control of the system is transferred to a system controller 286. The system controller administers a number of data stores

296, 292, and 290. Some of the data stores, such as the data store 296 further includes subordinate data stores 294. This subordinate data store 294 will include reporting means, in one embodiment. In another embodiment, the subordinate data store 294 includes information about a single customer, including sensitive information such as personally identifiable information. As such the subordinate data store 294 is maintained as a separate entity from the primary data store 296.

[107] As shown in Fig. 13, the advertiser and retailer data synchronizer 288 operates on the retailer and advertiser data stores 290, 292. The retailer data store 290 includes retailer assigned codes 300. The advertiser data store 292 includes advertiser assigned codes 302. The codes 300, 302 are used to cross link transactions including voucher code distribution 304, CPSS cross merchandising 306 and CPSS alphabet symbol linking 308, which are described below

[108] The main data store 296 is controlled by the system controller 286 and it includes tools to generate voucher codes 310, set access preferences 312. Further the main data store 296 includes the ability to access shopper profiles 314, shopper voucher codes 316 and for setting of customer access to CPSS link 318. The CPSS system is shown in detail in further figures and described below.

*Example Interfaces*

[109] Turning to Fig. 14, depicted therein is a sample interface 320 for a user of the system. The interface includes an action area 322 where the end user can select actions such as reviewing the points system, shopping for items, and other actions. The interface 320 also includes the end user's identifier 324 the process for which was discussed above. The interface 320 also allows the end user to log in to other areas of the system 326, such as if the end user also wanted to manage their own promotions. The interface 320 also allows the end user to redeem a winning code 328.

[110] Fig. 15 depicts two sample user interfaces. A first account setup interface 346 allows the end user to select an action 330 such as to log in or register. Presuming that the user has not previously registered with the system, the interface 346 allows the user to provide a name 332, email address 334, phone number 336, agree to terms and

conditions 338 and provide a date of birth 340. Once the information is filled in a user id is generated 342 and the end user can proceed in the system.

[111] A second interface is shown in Fig. 15, the second interface being for purchasing a discounted item 348. As part of the purchase, the end user may select the gift option 344, instead of buying the item for themselves.

[112] A view of an interface 350 of the Category Participating Sale System (CPSS) is shown in Fig. 16. The system allows registered advertisers 352 to view information such as the number of active advertisers and the number of active campaigns. In turn, registered advertisers 351 can review campaign performance by accessing their portal. For a particular campaign, the advertiser can assign voucher codes 354, can review the shopper database 356, and the number of vouchers that have been redeemed year to date 358. The user of the CPSS can also view the alphabet data links 360 and the relevant sections of the advertisers and retailers database 362. Other features include the cross merchandise units link 364, the customer preferences link 366, the ID system 368 and the options for setting of random values 370. Finally, the user can access the reward program 372.

[113] The details of a sample interface of the CPSS cross merchandising interface is shown in Fig. 17. The interface includes areas to select brands for each letter 380 and a listing of the number of active promotions 382 for each brand. The end user may select a number of actions 384 including selecting an alphabet block, assigning cash values, and others. Puzzles with the selected brands for cross merchandising appear in the preview 388 of a puzzle board 386. For example, in the embodiment shown in Fig. 17, the missing A of the 'AUTO' puzzle will be filled in with the A in the logo for Ace Hardware and so selecting that letter will result in cross merchandising. The end user of the interface can view and change the alphabet data links 381 and review information about cross merchandising 383.

[114] Example output of the cross merchandising is shown in Fig. 18. The game begins with a blank board 390. A number of randomly selected letters are filled in at step 392. The end user select letters during game play 394. Selecting the A during stage 396 results in receiving a cross merchandising offer 400 from the retailer 402

associated with the specially designated letter. The end user also receives the main offer 398 such as the ability to redeem a discount either in store 404 or online 406.

[115] A second alternative gameplay interface is shown in Fig. 19. In this embodiment, game again begins with a blank board 390. A number of randomly selected letters are filled in at step 392. The end user select letters during game play 394. Selecting the A during stage 396 results in receiving a mystery offer 408 from an unknown retailer 410. As before, the end user also receives the main offer 398 such as the ability to redeem a discount either in store 404 or online 406.

[116] Another interface is shown in Fig. 20. This interface shows another view of the CPSS module showing the number of assigned voucher codes 412, the database of advertisers 414 and the number of special alphabet links 416 and the total cost of the vouchers 418.

[117] Fig. 21 depicts a flow chart showing managerial functions within the system. The system involves a retailer or advertiser logging in 420. Once the login information is verified, access to the data store 422 is granted. The data store and the system are in communication with point of sale locations 440 using a data network 424. The system communicates with the data store 422 to ensure that voucher codes 426 are distributed. The advertiser or retailer logged into the system can manage product categories 428 and which products to advertise 430 using the promotions. The products are assigned product codes 432. The redemption codes and the product codes are linked to product venues 436 and point of sale locations 440. Finally, the advertiser or retailer can add cross merchandising links 438 and also include advanced retailer crosstalk 442.

## *System Schematic*

[118] The interactions between the system components and the various functions of one embodiment of the system are shown in schematic format in Fig. 22. As shown in Fig. 22, the system controller 450 manages the functions of the CPSS and other functions.

[119] The first function of the controller 450 is the cross selling premium products in alphabet links database, using the cross merchandising within the gameplay board 452.

[120] In one embodiment, the system verifies the inventory database 454 before making such cross-merchandising available. The cross merchandising products are advertised by registered advertisers by use of cross linked voucher codes 456. The voucher codes and other data is stored in a data store 458, in one embodiment. The registered advertisers also specify which products in the inventory 454 are to be award goods at the point of sale terminals 460.

[121] Once the settings are finalized 460, the puzzle boards are generated and added to receipts or otherwise displayed at point of sale locations 462. Two alpha numeric code banks are created at point of sale locations 464, 466. At the point of sale a receipt is generated 468. For the first code bank 464, the display of a discount amount and the item is set 468. The player will solve the puzzle on a paper cash register receipt from the point of sale 466. The end user may also bypass the paper receipt and play the game on a smart device 470. The second data store 466 will display the amount and item puzzle 472. Regardless of how the player reaches the app, the player will need to login or register with the app to gain entry into the system 474. The player can choose preferences to control category and link rewards 476 once they are logged into the system. The reward points are managed by the CPSS controller 478. The player will also set up a unique identifier and login to register to play the game 480. The player may also redeem codes and link to preferences as well as the CPSS 482. The player may also send reward codes as gifts 484 and check out purchases that used a won discount or other redemption code 486.

*Sample Consumer Interaction*

[122] Fig. 23 depicts a sample consumer interaction with the system. In the depicted embodiment, the point of sale terminal location 462 is in communication with a data network 424. The data network 424 provides a connection to a system that comprises an implementation of the system as described above. The customer interaction starts at beginning state 40. The consumer completes a point of sale

transaction, receiving a paper or digital receipt game piece 468. The consumer then plays the game on a paper receipt 468 as shown in Fig. 2A or on a digital device 470 as shown in Fig. 2B.

[123] Regardless of how the game starts, the consumer will log in to the application 474 and begin game play 44. The player will choose letters 52 and eventually the puzzle will be revealed 64. The player will solve the puzzle 60. The player may redeem a code 114, in one embodiment of the system. The prize is awarded 70 when the puzzle is revealed if the player meets the winning conditions 66. If the player loses 74, the player has the option to play again 76 which can be confirmed at state 79 or discontinue playing 78, in which case the game ends 80.

[124] It is to be understood that the above description is intended to be illustrative, and not restrictive. For example, the above-described embodiments (and/or aspects thereof) may be used in combination with each other. In addition, many modifications may be made to adapt a particular situation or material to the teachings of the invention without departing from its scope. While the dimensions and types of materials described herein are intended to define the parameters of the invention, they are by no means limiting, but are instead exemplary embodiments. Many other embodiments will be apparent to those of skill in the art upon reviewing the above description. The scope of the invention should, therefore, be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled. In the appended claims, the terms “including” and “in which” are used as the plain-English equivalents of the terms “comprising” and “wherein.” Moreover, in the following claims, the terms “first,” “second,” and “third,” are used merely as labels, and are not intended to impose numerical requirements on their objects.

[125] The present methods can involve any or all of the steps or conditions discussed above in various combinations, as desired. Accordingly, it will be readily apparent to the skilled artisan that in some of the disclosed methods certain steps can be deleted or additional steps performed without affecting the viability of the methods.

[126] As will be understood by one skilled in the art, for any and all purposes, particularly in terms of providing a written description, all ranges disclosed herein also

encompass any and all possible subranges and combinations of subranges thereof. Any listed range can be easily recognized as sufficiently describing and enabling the same range being broken down into at least equal halves, thirds, quarters, fifths, tenths, etc. As a non-limiting example, each range discussed herein can be readily broken down into a lower third, middle third and upper third, etc. As will also be understood by one skilled in the art all language such as "up to," "at least," "greater than," "less than," "more than" and the like include the number recited and refer to ranges which can be subsequently broken down into subranges as discussed above. In the same manner, all ratios disclosed herein also include all subratios falling within the broader ratio.

[127] One skilled in the art will also readily recognize that where members are grouped together in a common manner, such as in a Markush group, the present invention encompasses not only the entire group listed as a whole, but each member of the group individually and all possible subgroups of the main group. Accordingly, for all purposes, the present invention encompasses not only the main group, but also the main group absent one or more of the group members. The present invention also envisages the explicit exclusion of one or more of any of the group members in the claimed invention.

## Appendix A: Source Code for Web-based Embodiment

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<!-- GOOD FOR SEO -->
<meta name="description" content="">
<meta name="keywords" content="">
<meta name="copyright" content="Copyright © 2011 YourWebsite.com">
<meta name="author" content="Your Name">
<meta name="designer" content="Your Name">
<meta name="email" content="contact@email.com">
<meta name="robots" content="index, follow">
<meta name="googlebot" content="index, follow">
<meta name="revisit-after" content="1 Day">
<meta name="rating" content="General">
<meta name="distribution" content="Global">
<!-- GOOGLE VERIFY -->
<meta name="google-site-verification" content="XXX-XXX-XXX" />
<!-- ICON FAVI -->
<link rel="icon" type="image/x-icon" href="images/favicon.png">
<link rel="shortcut icon" type="image/x-icon" href="images/favicon.png">
<!-- FACEBOOK META SHARE -->
<meta property="og:title" content="" />
<meta property="og:type" content="website" />
<meta property="og:url" content="http://yourwebsite.com" />
<meta property="og:image"
        content="http://yourwebsite.com/images/facebookimage.png" />
<meta property="og:site_name" content="" />
<meta property="og:description" content="" />
<!-- TITLE OF WEBSITE -->
<title>Brandoodles on HTML</title>
<!-- WEBSITE SCRIPT & SOURCE -->
<link rel="stylesheet" href="css/reset.css">
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/jquery-ui-1.8.21.custom.css">
<script src="js/jquery.1.7.2.min.js" type="text/javascript"></script>
<script src="js/jquery-ui-1.8.21.custom.min.js"
type="text/javascript"></script>
<script src="js/jquery.global.js" type="text/javascript"></script>
<script src="js/test.js?1363119143" type="text/javascript"></script>
<script src="greensock/minified/TweenMax.min.js"
type="text/javascript"></script>
<script type="text/javascript" src="js/jquery.timer.js"></script>
<script type="text/javascript" src="js/timer.js"></script>
<script>
    $(function() {
        var options = {
                       letterWidth: 40,
            guess: 7,
            hint: 1,
            data:
{"1_1":{"text":"","solution":"Y","x":1,"y":1},"1_2":{"text":"o","solution":"o
","x":2,"y":1},"1_3":{"text":"o","solution":"o","x":3,"y":1},"1_4":{"text":"-
","solution":"-
","x":4,"y":1},"1_5":{"text":"H","solution":"H","x":5,"y":1},"1_6":{"text":"o
","solution":"o","x":6,"y":1},"1_7":{"text":"o","solution":"o","x":7,"y":1},"
2_1":{"text":"","solution":"C","x":1,"y":2},"2_2":{"text":"h","solution":"h",
"x":2,"y":2},"2_3":{"text":"o","solution":"o","x":3,"y":2},"2_4":{"text":"","
solution":"c","x":4,"y":2},"2_5":{"text":"o","solution":"o","x":5,"y":2},"2_6
":{"text":"","solution":"l","x":6,"y":2},"2_7":{"text":"a","solution":"a","x"
:7,"y":2},"2_8":{"text":"t","solution":"t","x":8,"y":2},"2_9":{"text":"e","so
lution":"e","x":9,"y":2},"3_1":{"text":"D","solution":"D","x":1,"y":3},"3_2":
{"text":"r","solution":"r","x":2,"y":3},"3_3":{"text":"","solution":"i","x":3
,"y":3},"3_4":{"text":"","solution":"n","x":4,"y":3},"3_5":{"text":"k","solut
ion":"k","x":5,"y":3}}        };
        $('#branArea').Brandle(options);
    });
        function replay() {
                //$('#branArea').html('');
                var options = {
                                letterWidth: 40,
                    guess: 7,
                    hint: 1,
                    data:
{"1_1":{"text":"V","solution":"V","x":1,"y":1},"1_2":{"text":"i","solution":"
i","x":2,"y":1},"1_3":{"text":"v","solution":"v","x":3,"y":1},"1_4":{"text":"
","solution":"a","x":4,"y":1},"1_5":{"text":"","solution":"r","x":5,"y":1},"1
_6":{"text":"i","solution":"i","x":6,"y":1},"1_7":{"text":"n","solution":"n",
"x":7,"y":1},"2_1":{"text":"","solution":"C","x":1,"y":2},"2_2":{"text":"","s
olution":"a","x":2,"y":2},"2_3":{"text":"f","solution":"f","x":3,"y":2},"2_4"
:{"text":"f","solution":"f","x":4,"y":2},"2_5":{"text":"","solution":"e","x":
5,"y":2},"2_6":{"text":"i","solution":"i","x":6,"y":2},"2_7":{"text":"n","sol
ution":"n","x":7,"y":2},"2_8":{"text":"","solution":"e","x":8,"y":2},"3_1":{"
text":"P","solution":"P","x":1,"y":3},"3_2":{"text":"i","solution":"i","x":2,
"y":3},"3_3":{"text":"","solution":"l","x":3,"y":3},"3_4":{"text":"","solutio
n":"l","x":4,"y":3},"3_5":{"text":"s","solution":"s","x":5,"y":3}}
};
                $('#branArea').Brandle(options);
                //clock.resetCountdown();
        }

        function showError() {
        var failure = document.getElementById("failure");
        TweenLite.fromTo(failure, 1, {top: 0, left: 210}, {top: 80, left:
210, ease: Back.easeInOut, onComplete: closeError});
        }

        function showSuccess(num) {
                var success = document.getElementById("success");
        success.innerHTML = "<p style='font-size: 40px;padding-
top:100px;padding-left:50px;font-family: Arial'>There are "+num+" letter in
the puzzle...</p>";
        TweenLite.fromTo(success, 1, {top: 0, left: 210}, {top: 80, left:
210, ease: Back.easeInOut, onComplete: closeSuccess});
        }

        function closeError() {
                var failure = document.getElementById("failure");
        TweenLite.to(failure, 1, {top: -999, left: 210, ease:
Back.easeInOut});
        }

        function closeSuccess() {
                var success = document.getElementById("success");
        TweenLite.to(success, 1, {top: -999, left: 210, ease:
Back.easeInOut});
        }

    function showHint() {
        var hint = document.getElementById("hint");
        TweenLite.fromTo(hint, 1, {top: 0, left: 210}, {top: 80, left: 210,
ease: Back.easeInOut, onComplete: closePopup});
    }
    function closePopup() {
       TweenLite.to('.popup', 1, {top: -999, left: 210, ease:
Back.easeInOut});
    }
    </script>
</head>
<body>
        <!-- CODE WILL BE HERE -->
        <div class="owrapper">
                <div id="branArea">
                        <div class="nav">
                                <div class="logo">
                                        <img src="images/logon.png" />
                                </div>
                                <div class="buttons">
                                        <!-- div class="puzzleinfo">
                                                <p><strong>Category:</strong>
Food</p>

                                                <p><strong>Word in
puzzle:</strong> 4</p>
                                        </div -->
                                        <div class="countdown">
                                                <p class="desc">Time</p>
                                                <span class="count"
id="countdown">05:00:00</span>
                                        </div>
                                        <div class="guess">
                                                <p class="text">Guesses</p>
                                                <p class="guess-count">5</p>
                                        </div>
                                        <div class="tag" id="info">
                                                <div style="margin-top: 15px">Go
go<br />go</div>
                                        </div>
                                </div>
                        </div>
                </div>
                <div id="failure" class="failure"
                        style="background: url(images/failed_popup.png) no-
repeat; width: 570px; height: 320px; display: block; position: absolute; top:
-9999px;"></div>
                <div id="success" class="success"
                        style="background: url(images/success_popup.png) no-
repeat; width: 570px; height: 320px; display: block; position: absolute; top:
-9999px;"></div>
                <div id="hint" class="hint popup"
```

```
                        style="background: url(images/success_popup.png) no-
repeat; width: 570px; height: 320px; display: block; position: absolute; top:
-9999px;">You
                        can only use 1 hint per game...</div>
                <div class="bshint">DO NOT REPEAT BUTTONS ALREADY PLAYED</div>
        </div>
        <div class="footer">
                <div class="lpart">
                        <img src="images/25-logo.png" />
                </div>
                <div class="rpart">
                        <img src="images/belcom-logo.png" />
                </div>
                <div class="mpart">
                        <div class="mininav">
                                <ul>
                                        <li><a href="">HOME</a></li>
                                        <li><a href="">ABOUT US</a></li>
                                        <li><a href="">RULES</a></li>
                                        <li><a href="">TERMS &
CONDITIONS</a></li>

                                        <li><a href="">REDEMPTION</a></li>
                                        <li><a href="">PRIVACY POLICY</a></li>
                                        <li><a href="">FAQ</a></li>
                                        <li><a href="">CONTACT US</a></li>
                                </ul>
                        </div>
                        <p>
                                Copyright &copy; 1991 - 2013. Belcom Corp.
Brandoodles is a
                                registered Trademark of Belcom Corp. All Rights
Reserved.<br />
                                Brandoodles.com consists of an independent
rewards promotional
                                program. Belcom Corp and Brandoodles.com<br />
are not affiliated
                                with any of the branded products or retailers
listed or stated in
                                the contents herein or any other<br /> Belcom
Corp programs.
                                Trademarks, service marks, logos, and/or domain
names (including,
                                without limitation,<br /> individual names of
products and
                                retailers) are the property of their respective
owners, having no
                                association<br /> with or make any endorsement
of the products or
                                services provided by Belcom Corp. or
Brandoodles.com.<br /> If you
                                have any questions regarding contents please
contact us at: <a

        href="mailto:info@belcomcorp.com">info@belcomcorp.com</a>.
                        </p>
                </div>
        </div>
```

```
</body>
</html>
```

## Appendix B: Random String Generator Source

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Random alphanumeric string generator</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
        <meta name="description" content="Generate random alphanumeric strings
of numbers, letters and symbols that can be used to create coupon codes,
passwords and other unique identifiers." />
        <meta name="keywords" content="alphanumeric strings, numbers, letters,
symbols, true random numbers" />
        <link rel="shortcut icon" href="images/favicon.ico?v4" />
        <base href="http://www.sweepjudge.com/" />
    <link href="styles.css" rel="stylesheet" type="text/css" />
    <link rel="stylesheet" type="text/css" href="css/styles.css?v4" />


        <script src="includes/prototype.js" type="text/javascript"></script>
        <script src="includes/campus.common.js"
type="text/javascript"></script>
        <script src="includes/login utils.js" type="text/javascript"></script>
        <script src="includes/random.js" type="text/javascript"></script>
</head>
<body>
        <!--Start header-->
        <div id="header">
                <div id="header_content">
                        <a href="/" id="logo"><img src="img/logo-new.png"
alt="Third party drawing service"/></a>
                        <!--Start login-logout-->
                                <NOBR><a class="login_logout"
href="login.htm"><span
class="stretch_button"><span><span>login</span></span></span></a></NOBR>
                        <!--End login-logout-->
                        <div class="clear"></div>
                                <table align="left" cellpadding="0"
cellspacing="0" style="margin-left:6px">
                                <tr>
                                <td>

                                <!--Start main-menu-->
                                                <ul id="main_menu">
                                                  <li class="first"><span><a
href="index.htm">Home</a></li><li><span><a
href="third party drawings.htm">Third Party Drawings</a></li><li><span><a
href="public records.htm">Public Records</a></li><li><span><a
href="free tools.htm">Free Randomizer Tools</a></li><li class="last"><span><a
href="blog.htm">Blog</a></li>
                                                </ul>
                                <!--Start end main-menu-->
                                </td>
                                </tr>
                                </table>
                        <div class="clear"></div>
                </div>
        </div>
        <!--End header-->
        <!--Start main area-->
        <div id="mainarea">
                <!--Start content-->
                <div id="content">
                        <!--Start center-->
                        <div class="center-top_angel">
                                <div class="center-bottom_angel">
                                        <div class="center">
                                                <!--Start drawing step4-->
                                                <div
class="drawing_step4_pages">
                                                        <h1>String
Generator</h1>
                                                        <table width="100%">
                                                        <tr>
                <tr>
                        <td   colspan="3" style="padding-top:3px;padding-
right:200px;background-image:url(images/random/string-image.jpg);background-
position:top right;background-repeat:no-repeat">

                                <p>The form below is a random string generator,
which can be utilized to generate a series of coupon codes, unique passwords
and any other random alphanumeric strings. By entering &quot;L&quot; for
Letters, &quot;N&quot; for Numbers and &quot;S&quot; for Symbols, the string
generator can create strings that are more than 10 characters in length. This
is one of the only random string generators that incorporate the optional use
of symbols to truly make it a complete solution for all of your string needs.
<BR><BR>
To begin generating random strings, simply enter how many Numbers, Letter and
Symbols you would like your string to consist of and select how many strings
you would like to generate
</p>
                        </td>
                </tr>
                <tr>
                        <td   colspan="3" height="30">
                          <div class="border_div"></div>
                        </td>
                </tr>
                <tr>
                        <td style="" colspan="3">
                                <h3>String Generator - Form</h3>
                                <table width="100%" class="steps-table">
                                        <tr>
                                                <td width="80">Step 1 -</td>
                                                <td>
                                                        Pattern: <input
size="10" maxlength="16" type="text" value="LL-NNNN" id="pat" /> (Use: N to
insert random digit, L-to insert random letter)
                                                </td>
                                        </tr>
                                        <tr>
                                                <td>Step 2 -</td>
                                                <td>
                                                        Number of records:
<input type="text" name="num" value="10" size="6" maxlength="12" id="num" />
                                                </td>
                                </tr>

                                <tr>
                                        <td>Final Step</td>
                                        <td>
                                                <input type="submit"
value="Generate" onclick="makeRequest()" />
                                                <input type="submit"
value="Reset form" onclick="resetForm()" />
                                        </td>
                                </tr>
                        </table>
                </td>
        </tr>
        <tr>
                <td   colspan="3" height="30">
                  <div class="border_div"></div>
                </td>
        </tr>
        <tr>
                <td style=""   valign="top" colspan="2">
                        <h3>String Generation - Results</h3>
                         <span id="results-stub"><br>To generate
results, please fill out the form above </span>
                        <pre id="results"></pre>
                </td>
                </td>
                <td width="500" style="border-left: 1px solid #D9D9D9;"
valign="top">
                                <div class="other-random-tools-small-
block">
                                  <h3>Free Randomizer Tools</h3>
                                                <p><a
href="lottery.htm">Lottery Quick Pick</a> is perhaps the Internet's most
popular with over 120 lotteries</p>
                                                <p><a
href="integer generator.htm">Number Generator</a> makes random numbers in
configurable intervals</p>
                                                <p><a
href="password generator.htm">Password Generator</a> makes secure passwords
for your Wi-Fi or that extra Gmail account</p>
                                                <p><a
href="string generator.htm">String Generator </a>makes random alphanumeric
strings and includes the ability incorporate symbols, much like you would
find when generating coupon codes for your Web site.</p>
                                </div>
                </td>
        </tr>
        </td>
    </tr>
</table>
</div>

                                        <!--End drawing step4-->
                                </div>
                        </div>
                </div>
                <!--End center-->
                </div>
                <!--End content-->
        </div>
        <!--End main area-->
        <!--Start footer-->
        <div id="footer">
                <div id="footer_content">
                        <ul id="footer_links">
                                <li>Third Party Drawings<br/>
                                        <a
href="third_party_drawings.htm">Overview</a>
                                </li>
                                <li>Public Records<br/>
                                        <a
href="public_records.htm">Results</a>
                                </li>
                                <li>Free Randomizer Tools<br/>
                                        <a href="integer_generator.htm">Number
Generator</a><br/>
                                        <a
href="password_generator.htm">Password Generator</a><br/>
                                        <a href="string_generator.htm">String
Generator</a><br/>
                                        <a href="lottery.htm">Lottery
Generator</a><br/>
                                </li>
                                <li>About SweepJudge<br/>
                                        <a href="contactus.htm">Contact
Us</a><br/>
                                        <a href="faq.htm">FAQ's</a><br/>
                                        <a
href="third_party_drawings_tac.htm">Terms &#38; Conditions</a><br/>
                                        <a href="policy.htm">Privacy
Policy</a><br/>
                                </li>
                                <li>Partners<br/>
                                        <a target="_blank"
href="http://www.cityopoly.com">Cityopoly</a><br/>
                                        <a target="_blank"
href="http://www.brollytime.com">Rain umbrellas</a><br/>
                                </li>
                        </ul>
                        <a href="/" id="footer_logo"><img
src="img/footer_logo.png" alt="Random number service"/></a>
                        <div class="clear"></div>
                        <p class="copyright">&#169; 2013 SweepJudge.com  All
Rights Reserved</p>
                </div>
        </div>
        <!--End footer-->

                <script type="text/javascript">
                 var AJAXBaseURL = 'http://www.sweepjudge.com';
                 var AJAXSessionParam =
'&PHPSESSID=uj82aiq4u718lvpipck20hg627';
                 var AJAXSessionID = 'uj82aiq4u718lvpipck20hg627';
                </script>
                <div id="campus_popup_overlay"></div>

<div id="campus-custom-popup" style="display:none">
 <div id="campus-custom-popup-html" style="float:left;width:92%">
 </div>
 <div style="float:right;width:25px;">
  <a href="javascript:;" id="campus-custom-popup-close-btn"
onclick="closeCustomModalDialog();return
false;"style="background:none;padding:5px; padding-right:10px;"><img
src="http://www.sweepjudge.com/images/close.gif" border="0"></a>
 </div>
</div>
<div id="campus-custom-tooltip"  style="display:none">
   <div id="campus-custom-tooltip-html"></div>
</div>
<script src="includes/ut.util.js" type="text/javascript"></script>
<script type="text/javascript">

  var _gaq = _gaq || [];
  _gaq.push(['_setAccount', 'UA-37565783-1']);
  _gaq.push(['_trackPageview']);

  (function() {
    var ga = document.createElement('script'); ga.type = 'text/javascript';
ga.async = true;
    ga.src = ('https:' == document.location.protocol ? 'https://ssl' :
'http://www') + '.google-analytics.com/ga.js';
    var s = document.getElementsByTagName('script')[0];
s.parentNode.insertBefore(ga, s);
  })();

</script>
        </body>
</html>
<script src="includes/strings.random.js" type="text/javascript"></script>
```

Strings.random.js

```
function resetForm()
{
        $('pat').value = 'NN-LLLL';
        $('len').value = 10;
}
function makeRequest()
{
        var num = $('num').value;
        var pat = $('pat').value;
        var code = '';
        new
Ajax.Request('/ajax/strings.php?pat='+pat+'&num='+num+'&contactCheck='+code,
        {
                method:'get',
                onSuccess: function(transport)
                {
                        var response = transport.responseText || "Wrong
parameters";
                        $('results-stub').hide();
                        CMPS_SetBlinkingText($('results'),response);
                },
                onFailure: function()
                {
                        alert('Error! Something went wrong...')
                }
        });
}
```

## Appendix C: Algorithms

```
from django.db import models

from Brandoodles.products.forms import DistributionFormField

class DistributionField(models.Field):

    __metaclass__ = models.SubfieldBase

    def __init__(self, *args, **kwargs):
        super(DistributionField, self).__init__(*args, **kwargs)

    def get_internal_type(self):
        return "CharField"

    def to_python(self, value):
        if isinstance(value, list):
            return value

        if not value:
            return []

        return map(int, value.split(','))

    def get_prep_value(self, value):
        return ','.join(map(unicode, value))

    def formfield(self, **kwargs):
        defaults = {'form_class': DistributionFormField}
        defaults.update(kwargs)

        return super(DistributionField, self).formfield(**defaults)

from django.contrib import admin
from django.shortcuts import render_to_response
from django.template import RequestContext
import django.forms as forms
from django.contrib import messages
from django.http import HttpResponseRedirect

import datetime

from Brandoodles.settings import distribution_values, initial_dist
from Brandoodles.products.models import Category, Product, Manufacturer
from Brandoodles.prizes.models import make_puzzle_start
from Brandoodles.products.forms import DistributionFormField


class CategoryAdmin(admin.ModelAdmin):
    list_display = ('code', 'name')

admin.site.register(Category, CategoryAdmin)
```

```
class ProductAdmin(admin.ModelAdmin):
    list_display = ('name', 'category', 'price')
    fields = (('name', 'manufacturer', 'picture'), ('category', 'price',
      'is_mystery'), 'distribution', 'retailers')
    filter_horizontal = ('retailers',)
    actions = ['generate_puzzles']

    class GenerateNewPuzzleForm(forms.Form):
        _selected_action = forms.CharField(widget=forms.MultipleHiddenInput)
        number = forms.IntegerField(initial=100)
        expiration = forms.DateField(initial=datetime.date.max)

    def generate_puzzles(self, request, queryset):
        if 'make puzzles' in request.POST:
            form = self.GenerateNewPuzzleForm(request.POST)
            if form.is_valid():
                number = form.cleaned_data['number']
                expiration = form.cleaned_data['expiration']

                for product in queryset:
                    weights = product.distribution
                    sum_weights = sum(weights)
                    number_to_generate = [int(number*w/sum_weights)
                                          for w in weights]
                    for i in range(len(weights)):
                        for j in range(number_to_generate[i]):
                            make_puzzle_start(product,
distribution_values[i], expiration)
                messages.success(request, "Puzzles created")

                return HttpResponseRedirect(request.get_full_path())
        else:
            form = self.GenerateNewPuzzleForm(initial={
                '_selected_action':
request.POST.getlist(admin.ACTION_CHECKBOX_NAME)})
        return render_to_response('new_puzzles.html',
                                  {'form': form,
                                   'path': request.get_full_path()},
                                  RequestContext(request))
    generate_puzzles.short_description = 'Add More Puzzles'

admin.site.register(Product, ProductAdmin)

admin.site.register(Manufacturer)
import django.forms as forms
import decimal

from Brandoodles.settings import distribution_values
from Brandoodles.settings import initial_dist

class DistributionWidget(forms.MultiWidget):
    def __init__(self, attrs=None):
        widgets = (forms.TextInput() for x in distribution_values)
        super(DistributionWidget, self).__init__(widgets, attrs)

    def format_output(self, rendered_widgets):
        out = ""
        for i in range(len(distribution_values)):
            out += unicode(int(distribution_values[i]*100)) + u"%: " +
rendered_widgets[i]
        return out

    def decompress(self, value):
        if isinstance(value, list) and value:
            return [int(x) if x else 0 for x in value]
        else:
            return [b for (_, b) in initial_dist]


class DistributionFormField(forms.MultiValueField):
    widget = DistributionWidget

    def __init__(self, *args, **kwargs):
        all_fields = (forms.IntegerField() for x in distribution_values)

        if "max_length" in kwargs:
            del kwargs["max_length"]
        super(DistributionFormField, self).__init__(all_fields, *args,
**kwargs)

    def clean(self, value):
        return [int(x) if x else 0 for x in value]

    def compress(self, data_list):
        return data_list
           forms.IntegerField(),
            forms.IntegerField(),
            )

        if "max_length" in kwargs:
            del kwargs["max_length"]
        super(DistributionFormField, self).__init__(all_fields, *args,
**kwargs)

    def compress(self, data_list):
        return list(data_list)
from django.db import models

from Brandoodles.retailers.models import Retailer
from Brandoodles.products.fields import DistributionField

class Category(models.Model):
    name = models.CharField(max_length=30)
    code = models.PositiveIntegerField(primary_key=True)

    def __unicode__(self):
        return self.name

    class Meta():
        verbose_name = "Category"
        verbose_name_plural = "Categories"

class Manufacturer(models.Model):
    name = models.CharField(max_length=30)
    logo = models.ImageField(upload_to='logo_images')

    def __unicode__(self):
        return self.name

class Product(models.Model):
    name = models.CharField(max_length=70)
    category = models.ForeignKey(Category)
    manufacturer = models.ForeignKey(Manufacturer)
    price = models.DecimalField(max_digits=12, decimal_places=2)
    picture = models.ImageField(upload_to='product_images')
    distribution = DistributionField(max_length=100, blank=True)
    is_mystery = models.BooleanField()
    retailers = models.ManyToManyField(Retailer, blank=True)

    def __unicode__(self):
        return self.name

from django.http import HttpResponse
from django.shortcuts import render

import datetime
import decimal

import Brandoodles.puzzle as brandoodles
from Brandoodles.prizes.models import Puzzle
from Brandoodles.products.models import Product

#The view which generates the main page, which primarily consists of the game
#interface.
def main_page(request):
    serial = request.session.get('prize_serial')
    game_data = request.session.get("current_puzzle")

    #Did they make a request?
    if request.method != 'POST':
        #Check whether there is an ongoing puzzle
        if "current_puzzle" in request.session:
            #Is the puzzle finished?
            if brandoodles.is_complete(game_data['letters_hidden']):
                #If so, display the completed puzzle and the prize
                return win_screen(request, game_data, serial)
            else:
                #Otherwise, display the game in progress
                return display_game_state(request, game_data, bool(serial))
        else:
            #No game in progress, so create a new one
            return new_game(request)
    else:
        #Did they request a new puzzle?
        if "newpuzzle" in request.POST or "current_puzzle" not in
request.session:
            return new_game(request)
        #Did they make a guess?
        elif "guess" in request.POST:
            #Adjust the state of the game appropriately
            guess = request.POST["guess"]
            if guess:
                new_letters = brandoodles.guess(game_data['letters_hidden'],
guess[0])
                game_data['letters_hidden'] = new_letters
                request.session["current_puzzle"] = game_data
                if brandoodles.is_complete(game_data['letters_hidden']):
                    return win_screen(request, game_data, serial)
                else:
                    return display_game_state(request, game_data,
bool(serial))
            else:
                return display_game_state(request, game_data, bool(serial))
        #Fallback on generating a new puzzle
        else:
            return new_game(request)


def new_game(request):
    #Get a random product from the product DB
    try:
        if not request.user.is_authenticated():
            raise(IndexError)
        newpuzzle =
Puzzle.objects.not_expired().filter(presented_on=None).select_related().order
_by('?')[0]
        product = newpuzzle.product
        category = product.category.name if not product.is_mystery else
"Mystery"
        solution = product.name
        starting_letters = newpuzzle.starting_letters
        newpuzzle.on_present()
        request.session["prize_serial"]=newpuzzle.serial
    #Fall back on random generation
    except IndexError:
        product = Product.objects.select_related().order_by('?')[0]
        category = product.category.name if not product.is_mystery else
"Mystery"
        solution = product.name
        starting_letters = brandoodles.make_random(solution)
        request.session["prize_serial"]=""
    newgame_data = {'solution': solution,
                    'category': category,
                    'letters_hidden': starting_letters}
    request.session["current_puzzle"] = newgame_data
    return display_game_state(request, newgame_data,
                              bool(request.session["prize_serial"]))


def display_game_state(request, game_data, prize_mode=False):
    return render(request, 'main_page.html',
                  {'puzzle':
brandoodles.render(game_data['solution'].upper(),
                                      game_data['letters_hidden']),
                   'words': len(game_data['solution'].split()),
                   'category': game_data['category'],
                   'demo': "Prize" if prize_mode else "Demo"})
```

```
def win_screen(request, game_data, serial=None):
    if serial:
        try:
            puzzle = Puzzle.objects.select_related().get(serial=serial)
        except:
            render(request, 'win_screen.html',
                        {'product': game_data['solution'],
                         'product_name': game_data['solution'].split()})
        else:
          puzzle.on_win()
          return render(request, 'win_screen.html',
                        {'product': puzzle.product,
                         'product_name': puzzle.product.name.split(),
                         'price': "%0.2f" % (puzzle.product.price),
                         'discount_percent': "%0.0f" % (100*puzzle.discount),
                         'discount_value': "%0.2f" %
(puzzle.discount_amount()),
                         'discounted_value': "%0.2f" %
puzzle.discounted_amount(),
                         'manufacturer': puzzle.product.manufacturer,
                         'retailers_left':
puzzle.product.retailers.all()[:2],
                         'retailers_right':
puzzle.product.retailers.all()[2:4],
                         'today': puzzle.won_on.strftime('%B %d, %Y'),
                         'expiry': puzzle.expiration.strftime('%x'),
                         'serial': serial})
    else:
        return render(request, 'win_screen.html',
                        {'product': game_data['solution'],
                         'product_name': game_data['solution'].split()})

"""
WSGI config for Brandoodles project.

This module contains the WSGI application used by Django's development server
and any production WSGI deployments. It should expose a module-level variable
named ``application``. Django's ``runserver`` and ``runfcgi`` commands
discover
this application via the ``WSGI_APPLICATION`` setting.

Usually you will have the standard Django WSGI application here, but it also
might make sense to replace the whole Django WSGI application with a custom
one
that later delegates to the Django one. For example, you could introduce WSGI
middleware here, or combine a Django application with an application of
another
framework.

"""
import os

# We defer to a DJANGO_SETTINGS_MODULE already in the environment. This
breaks
# if running multiple sites in the same mod_wsgi process. To fix this, use
# mod_wsgi daemon mode with each site in its own daemon process, or use
# os.environ["DJANGO_SETTINGS_MODULE"] = "Brandoodles.settings"
```

```
os.environ.setdefault("DJANGO_SETTINGS_MODULE", "Brandoodles.settings")

# This application object is used by any WSGI server configured to use this
# file. This includes Django's development server, if the WSGI_APPLICATION
# setting points here.
from django.core.wsgi import get_wsgi_application
application = get_wsgi_application()

# Apply WSGI middleware here.
# from helloworld.wsgi import HelloWorldApplication
# application = HelloWorldApplication(application)

from django.contrib import admin

from Brandoodles.retailers.models import Retailer

admin.site.register(Retailer)
from django.db import models

class Retailer(models.Model):
    name = models.CharField(max_length=30)
    address = models.CharField(max_length=50)
    city = models.CharField(max_length=60)
    state = models.CharField(max_length=30)
    zip_code = models.CharField(max_length=20)
    country = models.CharField(max_length=50)

    def __unicode__(self):
        return self.name
from django.contrib.auth.decorators import login_required
from django.shortcuts import render
from django.http import HttpResponseRedirect
from django.core.exceptions import ObjectDoesNotExist
from django.contrib import messages
from django.contrib.auth import login, authenticate
from django.contrib.auth.models import User
from django.contrib.auth.forms import PasswordChangeForm

from Brandoodles.players.models import Player
from Brandoodles.players.forms import FullUserForm, UserDataForm,
PlayerDataForm

#The view which generates a user profile, along with change forms
@login_required
def user_profile(request):
  user = request.user
  if request.method == 'POST':
    if "changeuserdata" in request.POST:
      userform = UserDataForm(request.POST, instance=user)
      if userform.is_valid():
      #modify user
        userform.save()
        messages.success(request, "Account information updated.")
      return render_profile(request, userform=userform)
    elif "changepassword" in request.POST:
      passwordform = PasswordChangeForm(user, request.POST)
      if passwordform.is_valid():
        #modify password
          passwordform.save()
          messages.success(request, "Password Changed.")
        return render_profile(request, passwordform=passwordform)
      elif "changeplayerdata" in request.POST:
        playerform = PlayerDataForm(request.POST)
        if playerform.is_valid():
        #modify player
          try:
            player = user.player
            playerform = PlayerDataForm(request.POST, instance=player)
            playerform.save()
            messages.success(request, "Player Info Changed.")
          except:
            playerform = PlayerDataForm(request.POST)
        return render_profile(request, playerform=playerform)
  return render_profile(request)

#The view for the registration form
def registration_form(request):
  if request.method == 'POST':
    userform = FullUserForm(request.POST)
    playerform = PlayerDataForm(request.POST)
    if userform.is_valid() and playerform.is_valid():
      user=userform.save()
      player=playerform.save(commit=False)
      player.user=user
      player.save()
      #Need to do it twice for the category preferences to take
      player.category_preferences =
playerform.cleaned_data['category_preferences']
      player.save()
      messages.success(request, 'Account created.')
      login(request, authenticate(username=user.username,
                                  password=request.POST['password']))
      return HttpResponseRedirect('/accounts/profile')
  else:
    userform = FullUserForm()
    playerform = PlayerDataForm()
  return render_registration(request, userform, playerform)

def render_profile(request, userform=None, playerform=None,
passwordform=None):
  user = request.user
  if not userform:
      userform = UserDataForm(initial={'first_name': user.first_name,
                                       'last_name': user.last_name,
                                       'email': user.email})
  if not passwordform:
      passwordform = PasswordChangeForm(user)
  try:
    if not playerform:
      player = user.player
      prefs = [x.code for x in player.category_preferences.all()]
      playerform = PlayerDataForm(initial={'phone': player.phone,
                                           'address': player.address,
                                           'city': player.city,
                                           'state': player.state,
                                           'zip_code': player.zip_code,
                                           'country': player.country,
                                           'dob': player.dob,
                                           'category_preferences': prefs})
    return render(request, 'registration/user_profile.html',
                  {'userform': userform,
                   'passwordform': passwordform,
                   'playerform': playerform})
  except ObjectDoesNotExist:
    return render(request, 'registration/user_profile.html',
                  {'userform': userform,
                   'passwordform': passwordform})

def render_registration(request, userform, playerform):
    return render(request, 'registration/user_registration.html',
                  {'userform': userform,
                   'playerform': playerform})
from django.db import models
from django.contrib import admin
from django.forms import CheckboxSelectMultiple

from Brandoodles.players.models import Player

class PlayerAdmin(admin.ModelAdmin):
    formfield_overrides = {
        models.ManyToManyField: {'widget': CheckboxSelectMultiple},
    }

admin.site.register(Player, PlayerAdmin)
from django import forms
from django.contrib.auth.models import User
from django.forms import CheckboxSelectMultiple

from Brandoodles.products.models import Category
from Brandoodles.players.models import Player

class FullUserForm(forms.ModelForm):
  def save(self, commit=True):
    user=super(FullUserForm, self).save(commit=False)
    user.set_password(self.cleaned_data["password"])
    if commit:
      user.save()
    return user

  class Meta:
    model = User
    fields = ['username', 'password', 'first_name', 'last_name', 'email']
    widgets = {'password': forms.PasswordInput}

class UserDataForm(forms.ModelForm):
  class Meta:
    model = User
    fields = ['first_name', 'last_name', 'email']

class PlayerDataForm(forms.ModelForm):
  class Meta:
    model = Player
    fields = ['phone',
              'address',
              'city',
              'state',
              'zip_code',
              'country',
              'dob',
              'category_preferences']
from django.db import models
from django.contrib.auth.models import User

import datetime

from Brandoodles.products.models import Category

class Player(models.Model):
    user = models.OneToOneField(User)
    phone = models.CharField(max_length=20, blank=True)
    address = models.CharField(max_length=50, blank=True)
    city = models.CharField(max_length=6, blank=True)
    state = models.CharField(max_length=30, blank=True)
    zip_code = models.CharField(max_length=20, blank=True)
    country = models.CharField(max_length=50, blank=True)
    dob = models.DateField("Date of Birth", help_text="Format is yyyy-mm-dd")
    category_preferences = models.ManyToManyField(Category,
                                              verbose_name="Preferences",
                                              blank=True,
                                              help_text="")

    def over_age(self, age=18):
        return age < abs(self.dob - datetime.date.today()) / 365

    def __unicode__(self):
        return self.user.__unicode__()

from django.contrib import admin

from Brandoodles.prizes.models import Puzzle

class PuzzleAdmin(admin.ModelAdmin):
    list_display = ('serial', 'product', 'discount', 'date_added',
'expiration',
                    'presented_on', 'won_on')

admin.site.register(Puzzle, PuzzleAdmin)

from django.db import models
from picklefield.fields import PickledObjectField

import random
import string
import datetime
from decimal import Decimal

from Brandoodles.products.models import Product
```

```
serial_digits = string.ascii_uppercase + string.digits
serial_length = 6

MAX_TRIES = 5 #Maximum tries to get a unique puzzle

class CouponExpiredManager(models.Manager):
    def expired(self):
        return self.filter(expiration__lt=datetime.date.today())
    def not_expired(self):
        return self.filter(expiration__gte=datetime.date.today())

class Puzzle(models.Model):
    starting_letters = models.CharField(max_length=100)

    product = models.ForeignKey(Product)

    serial = models.CharField(max_length=20, unique=True)
    discount = models.DecimalField(max_digits=5, decimal_places=4)
    date_added = models.DateField()
    expiration = models.DateField()

    presented_on = models.DateField(null=True, blank=True)
    won_on = models.DateField(null=True, blank=True)

    objects = CouponExpiredManager()

    def is_expired(self):
        return self.expiration < datetime.date.today()

    def discount_amount(self):
        return self.discount * self.product.price

    def discounted_amount(self):
        return (1-self.discount) * self.product.price

    def on_present(self):
        self.presented_on = datetime.date.today()
        self.save()

    def on_win(self):
        self.won_on = datetime.date.today()
        self.save()

    def __unicode__(self):
        return self.serial

#    class Meta:
        #This guarantees uniqueness but can cause errors
#        unique_together = ('starting_letters', 'product')

def make_puzzle_start(product, discount, expiration_date,
                      min_difficulty=0.5):
#Generate Unique Starting Puzzle
    soln = product.name.upper()
    letters_used = [x for x in set(soln) if not x.isspace()
                    and x not in string.punctuation]
    max_to_show = int(round((1-min_difficulty) * len(letters_used)))
    starting_puzzle = None
    tries = 0
    while not starting_puzzle:
        tries = tries + 1
        number_to_show = random.randint(0,max_to_show)
        starting_puzzle = ''.join(random.sample(letters_used,
number_to_show))
        if tries < MAX_TRIES and \
           Puzzle.objects.filter(starting_letters=starting_puzzle,
                                 product=product).exists():
            starting_puzzle = None

#Generate Unique Serial Code
    new_serial = ""
    todaysyear = str(datetime.date.today().year)
    while not new_serial:
        new_serial = todaysyear[-2:] + \
            ''.join([random.choice(serial_digits)
                     for i in range(0, serial_length)])
        if Puzzle.objects.filter(serial=new_serial).exists():
            new_serial = ""

#Add the puzzle to the database
    Puzzle.objects.create(starting_letters=starting_puzzle, product=product,
                          serial=new_serial, discount=discount,
                          date_added=datetime.date.today(),
                          expiration=expiration_date)
erial, discount=discount,
                          date_added=datetime.date.today(),
                          expiration=expiration_date)
# Django settings for Brandoodles project.
import os, decimal

def local_path(p):
  return os.path.join(os.path.dirname(__file__), p).replace('\\', '/')

distribution_values = map(lambda x: decimal.Decimal(x)/10, range(1, 10))
initial_dist = zip(distribution_values, (200,170,150,130,110,90,70,50,30))

DEBUG = True
TEMPLATE_DEBUG = DEBUG

ADMINS = (
    # ('Your Name', 'your_email@example.com'),
)

MANAGERS = ADMINS

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': local_path('brandoodles.db'),
        'USER': '',
        'PASSWORD': '',
        'HOST': '',
        'PORT': '',
    }
```

```
}

# Hosts/domain names that are valid for this site; required if DEBUG is False
# See https://docs.djangoproject.com/en/1.5/ref/settings/#allowed-hosts
ALLOWED_HOSTS = []

# Local time zone for this installation. Choices can be found here:
# http://en.wikipedia.org/wiki/List_of_tz_zones_by_name
# although not all choices may be available on all operating systems.
# In a Windows environment this must be set to your system time zone.
TIME_ZONE = 'America/Chicago'

# Language code for this installation. All choices can be found here:
# http://www.i18nguy.com/unicode/language-identifiers.html
LANGUAGE_CODE = 'en-us'

SITE_ID = 1

# If you set this to False, Django will make some optimizations so as not
# to load the internationalization machinery.
USE_I18N = True

# If you set this to False, Django will not format dates, numbers and
# calendars according to the current locale.
USE_L10N = True

# If you set this to False, Django will not use timezone-aware datetimes.
USE_TZ = True

# Absolute filesystem path to the directory that will hold user-uploaded
files.
# Example: "/var/www/example.com/media/"
MEDIA_ROOT = local_path('media')

# URL that handles the media served from MEDIA_ROOT. Make sure to use a
# trailing slash.
# Examples: "http://example.com/media/", "http://media.example.com/"
MEDIA_URL = '/media/'

# Absolute path to the directory static files should be collected to.
# Don't put anything in this directory yourself; store your static files
# in apps' "static/" subdirectories and in STATICFILES_DIRS.
# Example: "/var/www/example.com/static/"
STATIC_ROOT = local_path('static')

# URL prefix for static files.
# Example: "http://example.com/static/", "http://static.example.com/"
STATIC_URL = '/static/'

# Additional locations of static files
STATICFILES_DIRS = (
    # Put strings here, like "/home/html/static" or "C:/www/django/static".
    # Always use forward slashes, even on Windows.
    # Don't forget to use absolute paths, not relative paths.
)

# List of finder classes that know how to find static files in
# various locations.
STATICFILES_FINDERS = (
    'django.contrib.staticfiles.finders.FileSystemFinder',
    'django.contrib.staticfiles.finders.AppDirectoriesFinder',
#    'django.contrib.staticfiles.finders.DefaultStorageFinder',
)

# Make this unique, and don't share it with anybody.
SECRET_KEY = '2uewab13iau!)e+!5^ysq&6-2*w5*-a=jar--=7f$7igt20#eo'

# List of callables that know how to import templates from various sources.
TEMPLATE_LOADERS = (
    'django.template.loaders.filesystem.Loader',
    'django.template.loaders.app_directories.Loader',
#     'django.template.loaders.eggs.Loader',
)

MIDDLEWARE_CLASSES = (
    'django.middleware.common.CommonMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    # Uncomment the next line for simple clickjacking protection:
    # 'django.middleware.clickjacking.XFrameOptionsMiddleware',
)

ROOT_URLCONF = 'Brandoodles.urls'

# Python dotted path to the WSGI application used by Django's runserver.
WSGI_APPLICATION = 'Brandoodles.wsgi.application'

TEMPLATE_DIRS = (
    # Put strings here, like "/home/html/django_templates" or
"C:/www/django/templates".
    # Always use forward slashes, even on Windows.
    # Don't forget to use absolute paths, not relative paths.
    local_path('templates')
)

INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    # Uncomment the next line to enable the admin:
    'django.contrib.admin',
    # Uncomment the next line to enable admin documentation:
    # 'django.contrib.admindocs',
    'Brandoodles.products',
    'Brandoodles.prizes',
    'Brandoodles.retailers',
    'Brandoodles.players',
)
```

```
# A sample logging configuration. The only tangible logging
# performed by this configuration is to send an email to
# the site admins on every HTTP 500 error when DEBUG=False.
# See http://docs.djangoproject.com/en/dev/topics/logging for
# more details on how to customize your logging configuration.
LOGGING = {
    'version': 1,
    'disable_existing_loggers': False,
    'filters': {
        'require_debug_false': {
            '()': 'django.utils.log.RequireDebugFalse'
        }
    },
    'handlers': {
        'mail_admins': {
            'level': 'ERROR',
            'filters': ['require_debug_false'],
            'class': 'django.utils.log.AdminEmailHandler'
        }
    },
    'loggers': {
        'django.request': {
            'handlers': ['mail_admins'],
            'level': 'ERROR',
            'propagate': True,
        },
    }
}
from django.conf.urls import patterns, include, url
from django.contrib import admin
from django.conf import settings
from django.conf.urls.static import static

from Brandoodles.views import main_page
from Brandoodles.players.views import user_profile, registration_form

admin.autodiscover()

urlpatterns = patterns('',
    #Admin site
    url(r'^admin/', include(admin.site.urls)),

    #User auth pages
    url(r'^accounts/login', 'django.contrib.auth.views.login'),
    url(r'^accounts/logout', 'django.contrib.auth.views.logout_then_login'),
    url(r'^accounts/register', registration_form),
    url(r'^accounts/profile', user_profile),

    #Game-related pages
    url(r'^$', main_page),
) + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
import random
import textwrap
import string

puzzle_row_width = 16
puzzle_col_width = 4
```

```
def form(puzzle):
#String formatting for a puzzle
    rows = puzzle.split()  # Separate lines for each word
    while len(rows) > puzzle_col_width:
        rpairs = [' '.join(rows[i:i+2]) for i in range(len(rows)-1)]
        mrow = 0
        mlen = len(rpairs[0])
        for i in range(1, len(rpairs)):
            curlen = len(rpairs[i])
            if curlen < mlen:
                mlen = curlen
                mrow = i
        rows = rows[:mrow] + [rpairs[mrow]] + rows[mrow+2:]
    return rows

def make_random(solution, fraction_to_hide=0.75):
#Randomly generate a puzzle
    letters_used = [x for x in set(solution) if not x.isspace()
                    and x not in string.punctuation]
    number_to_hide = int(round(fraction_to_hide * len(letters_used)))
    starting_letters = ''.join(random.sample(letters_used, number_to_hide))
    return starting_letters.upper()

def is_complete(letters_hidden):
#Determine whether a puzzle is done
    return not letters_hidden

def guess(letters_hidden, g):
#Process a guess toward completing the puzzle
    g = g.upper()
    return letters_hidden.replace(g, '')

def render(solution, letters_hidden):
#Produce a string representation of a solution
    puzzle = solution
    for c in letters_hidden:
        puzzle = puzzle.replace(c, '_')
    return form(puzzle)
{% if user.is_authenticated %}
  <p>Logged in as: {{user.username}}.
    <a href="/accounts/logout">Logout</a>
    <a href="/accounts/profile">Profile</a></p>
{% else %}
  <p> You are not logged in.
    <a href="/accounts/login">Log In</a>
    <a href="/accounts/register">Register</a></p>
{% endif %}
<html>
<body>
{% if messages %}
<ul class="messages">
  {% for message in messages %}
    <li{% if message.tags %} class="{{message.tags }}"{% endif %}>{{ message
}}</li>
  {% endfor %}
</ul>
```

```
{% endif%}
User: {{ user.username }}
<hr width="100%">
<form method="post">{% csrf_token %}
  {{ userform.as_p }}
  <input type="submit" value="Change User Data" name="changeuserdata"/>
</form>
<hr width="100%">
<form method="post">{% csrf_token %}
  {{ passwordform.as_table }}
  <input type="submit" value="Change Password" name="changepassword"/>
</form>
{% if playerform %}
<hr width="100%">
  <form method="post">{% csrf_token %}
    {{ playerform.as_p }}
  <input type="submit" value="Change Player Profile"
name="changeplayerdata"/>
  </form>
{% endif %}
<hr width="100%">
<a href="/">Start Playing</a>
</body>
</html>
{% if form.errors %}
<p>Your username and password didn't match. Please try again.</p>
{% endif %}

<form method="post" action="{% url 'django.contrib.auth.views.login' %}">
{% csrf_token %}
<table>
<tr>
    <td>{{ form.username.label_tag }}</td>
    <td>{{ form.username }}</td>
</tr>
<tr>
    <td>{{ form.password.label_tag }}</td>
    <td>{{ form.password }}</td>
</tr>
</table>

<input type="submit" value="login" />
<input type="hidden" name="next" value="{{ next }}" />
</form>

<html>
<body>
{% if messages %}
<ul class="messages">
  {% for message in messages %}
    <li{% if message.tags %} class="{{message.tags }}"{% endif %}>{{ message
}}</li>
  {% endfor %}
</ul>
{% endif%}
<form method="post">{% csrf_token %}
  {{ userform.as_p }}
  {{ playerform.as_p }}
  <input type="submit" value="Create Account"/>
</form>
</body>
</html>
<html>
<head>
 <link rel="stylesheet" type="text/css" href="/media/css/coupon.css" />
</head>
<body>
 <center> You Won <br>
  {{ today }} <br> </center>
 <center>
 Because you played <br>
<img src="/media/brandoodles_logo.png" id="brandoodles_logo"/></center>
{% if user.is_authenticated and serial %}
  <center> Coupon ID: {{ serial }} </center>
  <div id="barcode"></div>
  <center> Redeem this coupon for <br>
    <strong>{{ discount_percent }}% </strong> <br>
    off the retail price of this item</center>
  <img id="product_logo" src="/media/{{ product.picture }}"/>
  <div id="product_name">
    <big><big>{% for l in product_name %}
    {{ l }} <br>
    {% endfor %}</big></big><br clear=all>
  </div>
  <hr width="100%">
  <center> Redeem at these establishments <br>
    for up to ${{ discount_value }} savings on this item. <br>
   This offer expires: <br>
   {{ expiry }}</center>
   <div id="retailers_section">
     <div id="retailers_left">
       {% for r in retailers_left %}
         {{ r.name }} <br>
         {{ r.city }}, {{ r.state }}, {{ r.zip_code }} <br><br><br>
       {% endfor %}
     </div>
     <div id="retailers_right">
       {% for r in retailers_right %}
         {{ r.name }} <br>
         {{ r.city }}, {{ r.state }}, {{ r.zip_code }} <br><br><br>
       {% endfor %}
     </div>
     <div id="retailers_center" align="center">
       <img align="center" width="20%" src="/media/{{ manufacturer.logo }}">
     </div>
   </div>
  <br clear=all>
  <br>
{% else %}
  <div id="product_name">
    <big><big>{% for l in product_name %}
    {{ l }} <br>
    {% endfor %}</big></big><br clear=all>
  </div>
```

```
{% endif %}

<form name="newpuzzle" method="post" action="">
  <input type="submit" name="newpuzzle" value="New Puzzle">
  {% csrf_token %}
</form>
{% include "login_info.html" %}
  <script type="text/javascript" src="media/jquery-1.3.2.min.js"></script>
  <script type="text/javascript" src="media/jquery-barcode.js"></script>
  <script type="text/javascript">
    $("#barcode").barcode("{{ serial }}", "code128")
  </script>
</body>
</html>

<html>
  <body>
    <form method="post" action="{{ path }}">{% csrf_token %}
      <table>
        {{ form.as_table }}
      </table>
      <p>
        <input type="hidden" name="action" value="generate_puzzles" />
        <input type="submit" name="make puzzles" value="Make Puzzles" />
      </p>
    </form>
  </body>
</html>
<html>

<body>
Mode: {{ demo }}<br>
Category: {{ category }}<br>
Words: {{ words }}<br>

<br>

{% for l in puzzle %}
{{ l }} <br>
{% endfor %}
<br>
<form name="guess" method="post" action="">

<input type="text" name="guess"/>
<input type="submit" value="Guess">
  {% csrf_token %}
</form>
<form name="newpuzzle" method="post" action="">
  <input type="submit" name="newpuzzle" value="New Puzzle">
  {% csrf_token %}
</form>
{% include "login_info.html" %}
</body>
</html>
```

Claims:

1. A method for promoting sales and increasing brand recognition comprising:

selecting by a promotion administrator product categories at a website administration portal hosted on an administration server;

updating by the administration server a retailer database and a product database;

selecting by the promotion administrator which brand will participate in a promotion at a retailer location and rewards associated with each promotion;

assigning by the administration server random alphanumeric codes to be used at point of sale terminals along with puzzles;

transmitting by the administration server to the retailer database promotion information including a set of codes and brand name puzzles to be printed on receipts;

selecting by one of the point of sale terminals, a brand name and alphanumeric code from the retailer database based on customer purchases;

printing by the one of the point of sale terminals a paper receipt which includes a list of purchases and a game piece wherein the game piece comprises the alphanumeric code and a selected puzzle;

downloading a game application on a device of a customer, wherein the game application is in communication with the administration server and the retailer database;

accepting by the game application the printed alphanumeric code from the paper receipt;

displaying on the device the selected puzzle retrieved from the retailer database;

wherein the customer enters letters into the device, thereby completing the selected puzzle during a required time period;

determining by the game application whether the customer completed the selected puzzle within the required time period by the game application comparing an entry of the customer with a solution to the selected puzzle;

Date Reçue/Date Received 2021-07-28

verifying by the application that the customer solved the selected puzzle; and

when it is determined that the customer solved the selected puzzle, retrieving by the application from the administration server a prize depending on the entries in the retailer database, the prize selected by a prize administrator, and presenting by the application the prize to the customer.

2. The method as recited in claim 1 wherein the displaying step further comprises selecting by the game application one or more random letters to display thereby guiding the customer in solving the selected puzzle.

3. The method as recited in claim 1 wherein the prize retrieved from the administration server comprises a cross-merchandising offer based on current inventory levels as determined by the administration server after referring to the retailer database.

4. The method as recited in claim 1 wherein the website administration portal comprises a dashboard wherein the promotion administrator selects the number of products and a brand alphabet.

Date Reçue/Date Received 2021-07-28

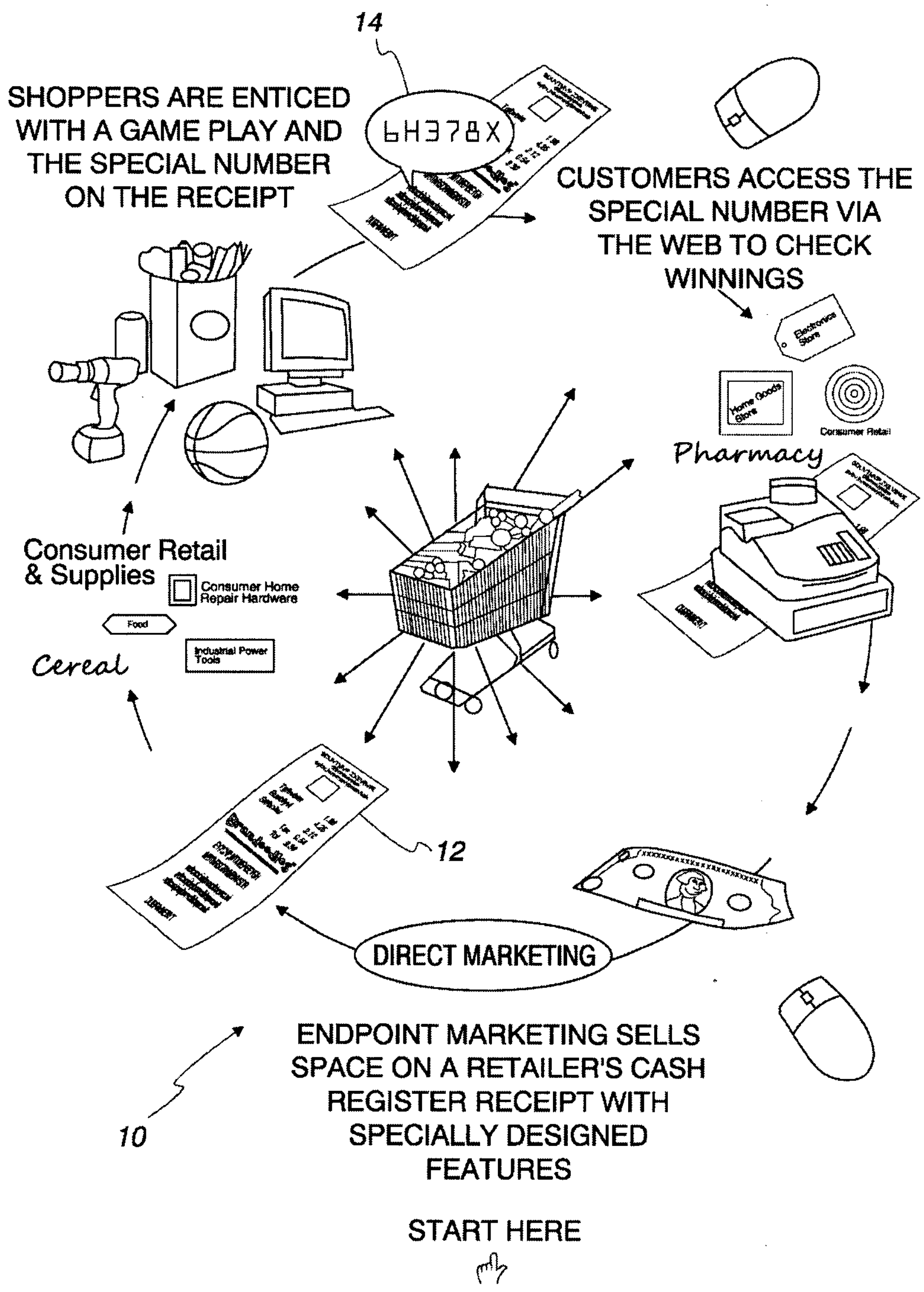

14
SHOPPERS ARE ENTICED WITH A GAME PLAY AND THE SPECIAL NUMBER ON THE RECEIPT
bH378X
CUSTOMERS ACCESS THE SPECIAL NUMBER VIA THE WEB TO CHECK WINNINGS
Electronics Store
Home Goods Store
Consumer Retail
Pharmacy
Consumer Retail & Supplies
Consumer Home Repair Hardware
Food
Industrial Power Tools
Cereal
12
DIRECT MARKETING
ENDPOINT MARKETING SELLS SPACE ON A RETAILER'S CASH REGISTER RECEIPT WITH SPECIALLY DESIGNED FEATURES
10
START HERE

Fig. 1

Replacement Sheet

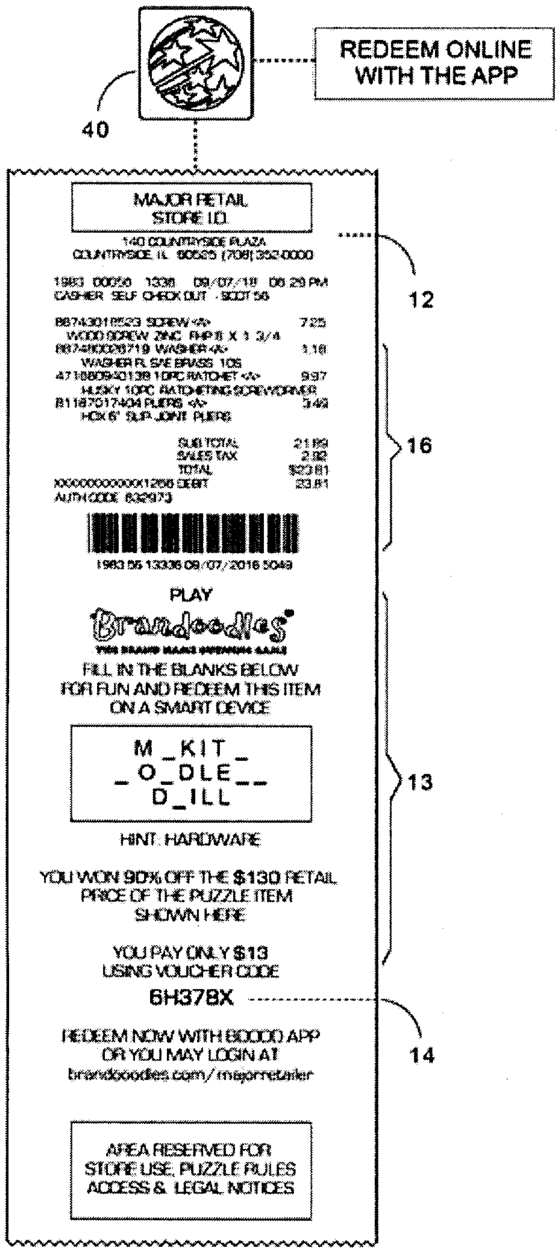

REDEEM ONLINE WITH THE APP
40
MAJOR RETAIL STORE I.D.
140 COUNTRYSIDE PLAZA
COUNTRYSIDE, IL 60525 (708) 352-0000
1983 00056 1336 09/07/18 06:29 PM
CASHIER SELF CHECK OUT - SCOT 56
88743018523 SCREW <A> 7.25
WOOD SCREW ZINC PHP 8 X 1 3/4
887410026719 WASHER <A> 1.18
WASHER FL SAE BRASS 10S
471880940138 10PC RATCHET <A> 9.97
HUSKY 10PC RATCHETING SCREWDRIVER
81187017404 PLIERS <A> 3.49
HDX 6" SLIP JOINT PLIERS
SUB TOTAL 21.89
SALES TAX 2.92
TOTAL $23.81
XXXXXXXXXXXX1266 DEBIT 23.81
AUTH CODE 632973
1983 56 13336 09/07/2016 5049
PLAY
Brandoodles
FILL IN THE BLANKS BELOW FOR FUN AND REDEEM THIS ITEM ON A SMART DEVICE
M_KIT_
_O_DLE__
D_ILL
HINT: HARDWARE
YOU WON 90% OFF THE $130 RETAIL PRICE OF THE PUZZLE ITEM SHOWN HERE
YOU PAY ONLY $13 USING VOUCHER CODE
6H37BX
REDEEM NOW WITH BOODO APP OR YOU MAY LOGIN AT brandoodles.com/majorretailer
AREA RESERVED FOR STORE USE, PUZZLE RULES ACCESS & LEGAL NOTICES
12
16
13
14

Fig. 2A

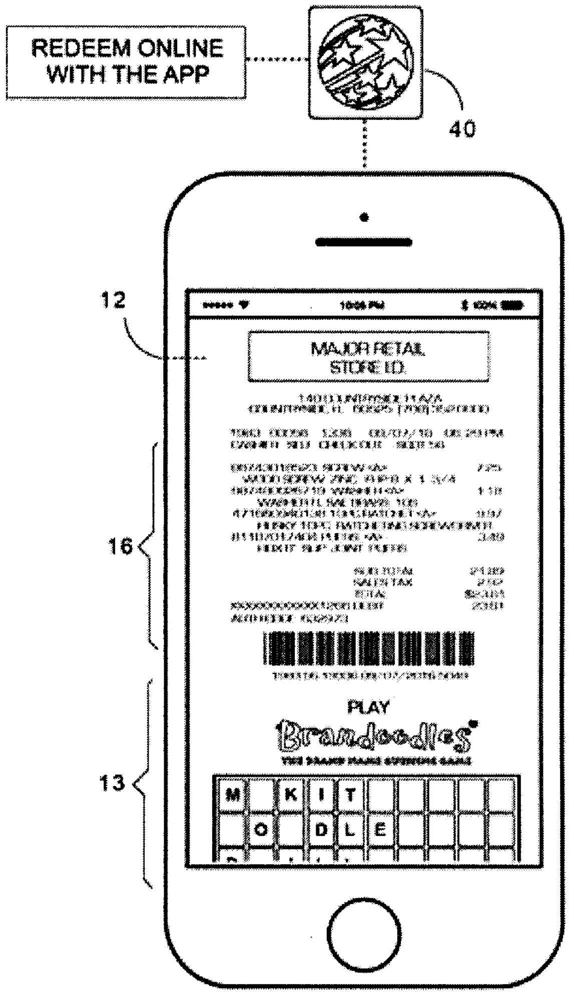


Fig. 2B

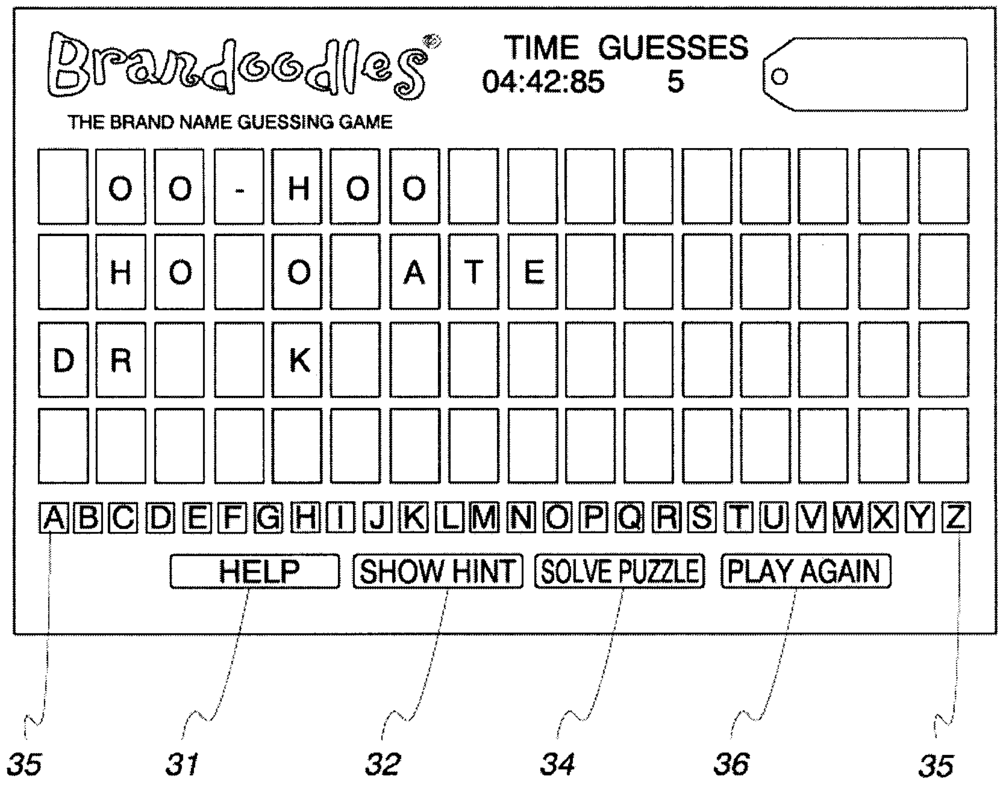

Brandoodles®
THE BRAND NAME GUESSING GAME
TIME GUESSES
04:42:85 5
O O - H O O
H O O A T E
D R K
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
HELP
SHOW HINT
SOLVE PUZZLE
PLAY AGAIN
35
31
32
34
36
35

Fig. 3

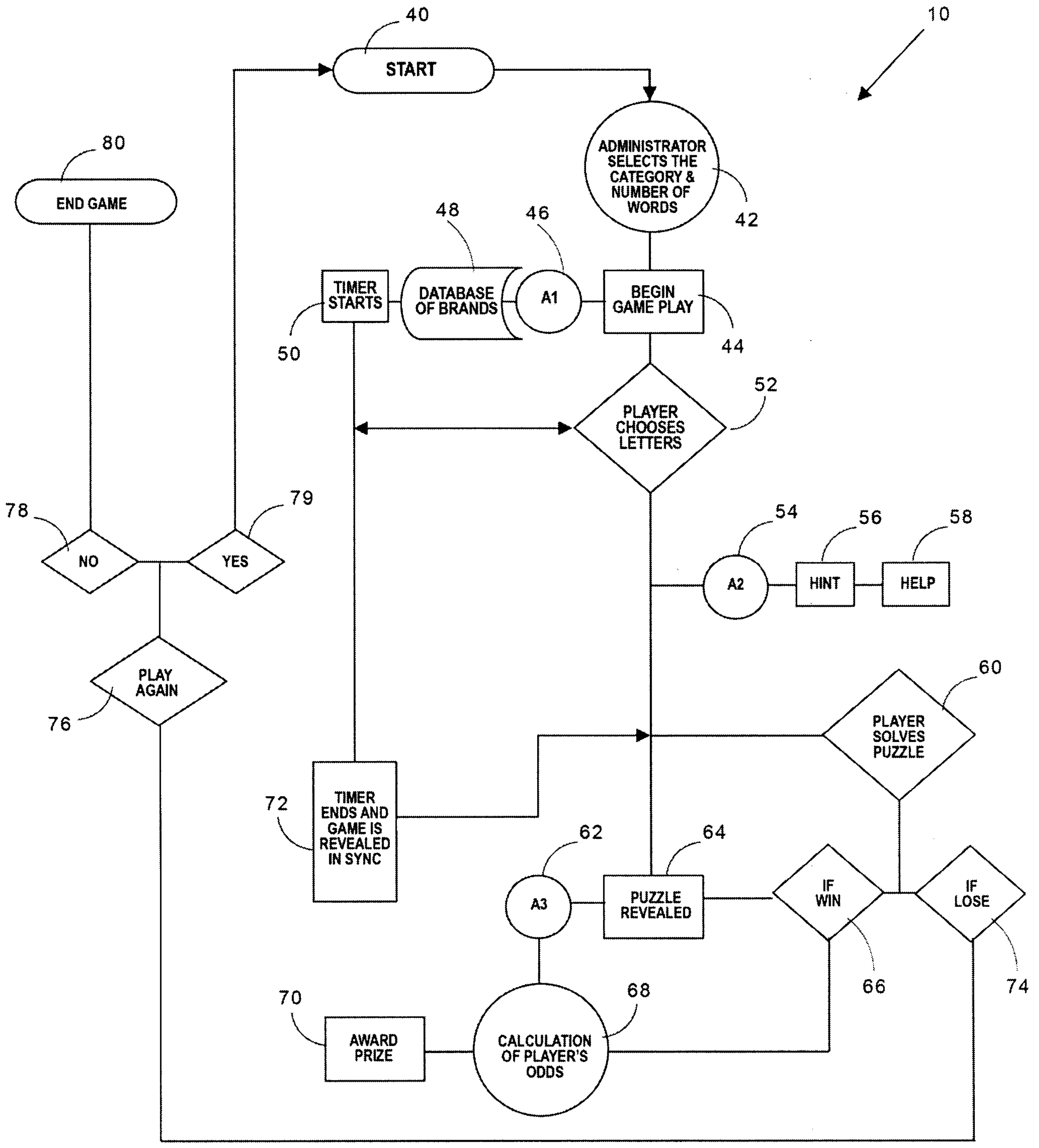

10
40
START
ADMINISTRATOR SELECTS THE CATEGORY & NUMBER OF WORDS
42
80
END GAME
48
46
TIMER STARTS
DATABASE OF BRANDS
A1
BEGIN GAME PLAY
50
44
52
PLAYER CHOOSES LETTERS
78
79
NO
YES
54
56
58
A2
HINT
HELP
PLAY AGAIN
76
60
PLAYER SOLVES PUZZLE
72
TIMER ENDS AND GAME IS REVEALED IN SYNC
62
64
A3
PUZZLE REVEALED
IF WIN
IF LOSE
66
74
70
68
AWARD PRIZE
CALCULATION OF PLAYER'S ODDS

Fig. 4

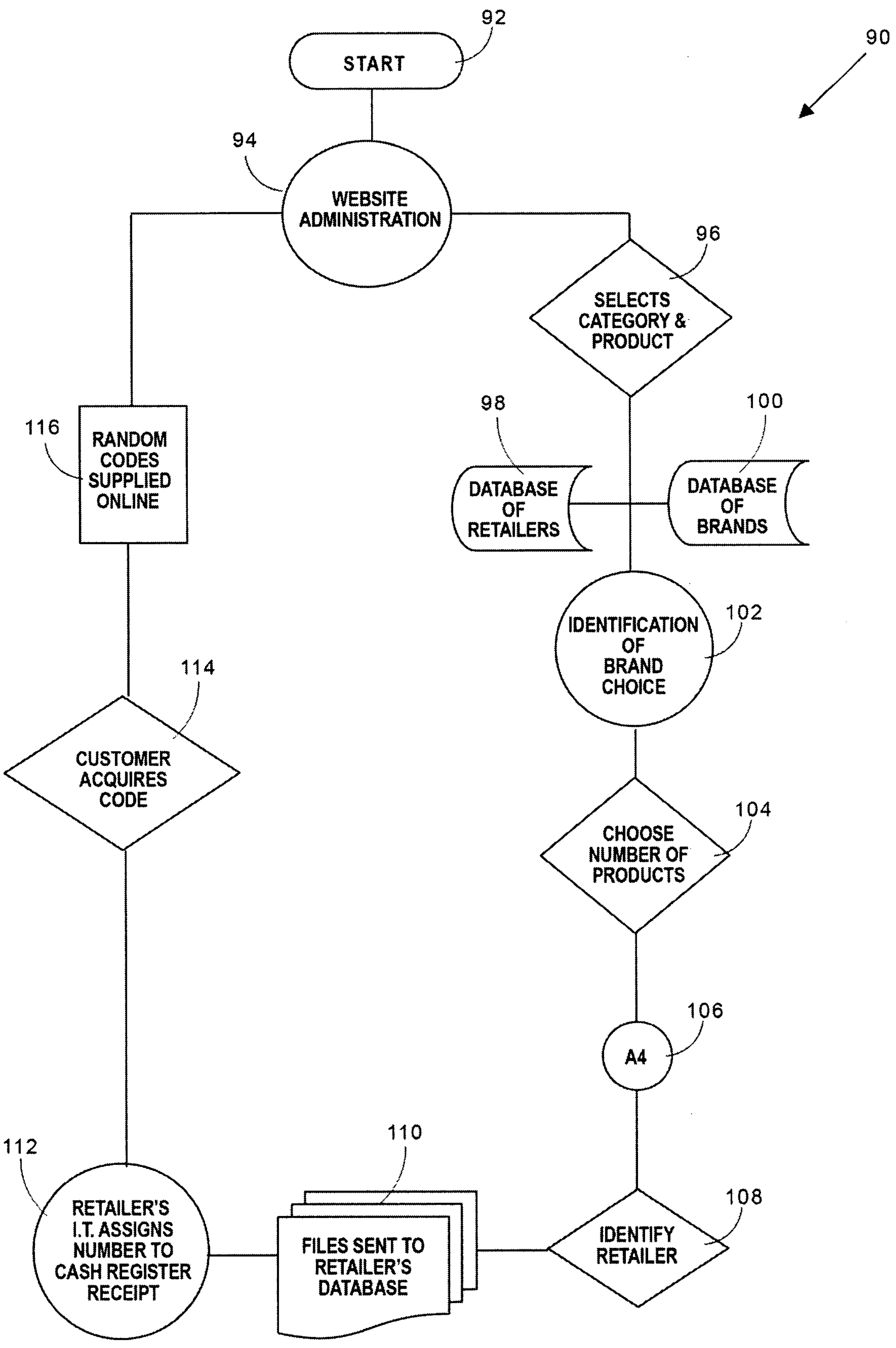

92
START
90
94
WEBSITE ADMINISTRATION
96
SELECTS CATEGORY & PRODUCT
98
DATABASE OF RETAILERS
100
DATABASE OF BRANDS
116
RANDOM CODES SUPPLIED ONLINE
102
IDENTIFICATION OF BRAND CHOICE
114
CUSTOMER ACQUIRES CODE
104
CHOOSE NUMBER OF PRODUCTS
106
A4
112
RETAILER'S I.T. ASSIGNS NUMBER TO CASH REGISTER RECEIPT
110
FILES SENT TO RETAILER'S DATABASE
108
IDENTIFY RETAILER

Fig. 5

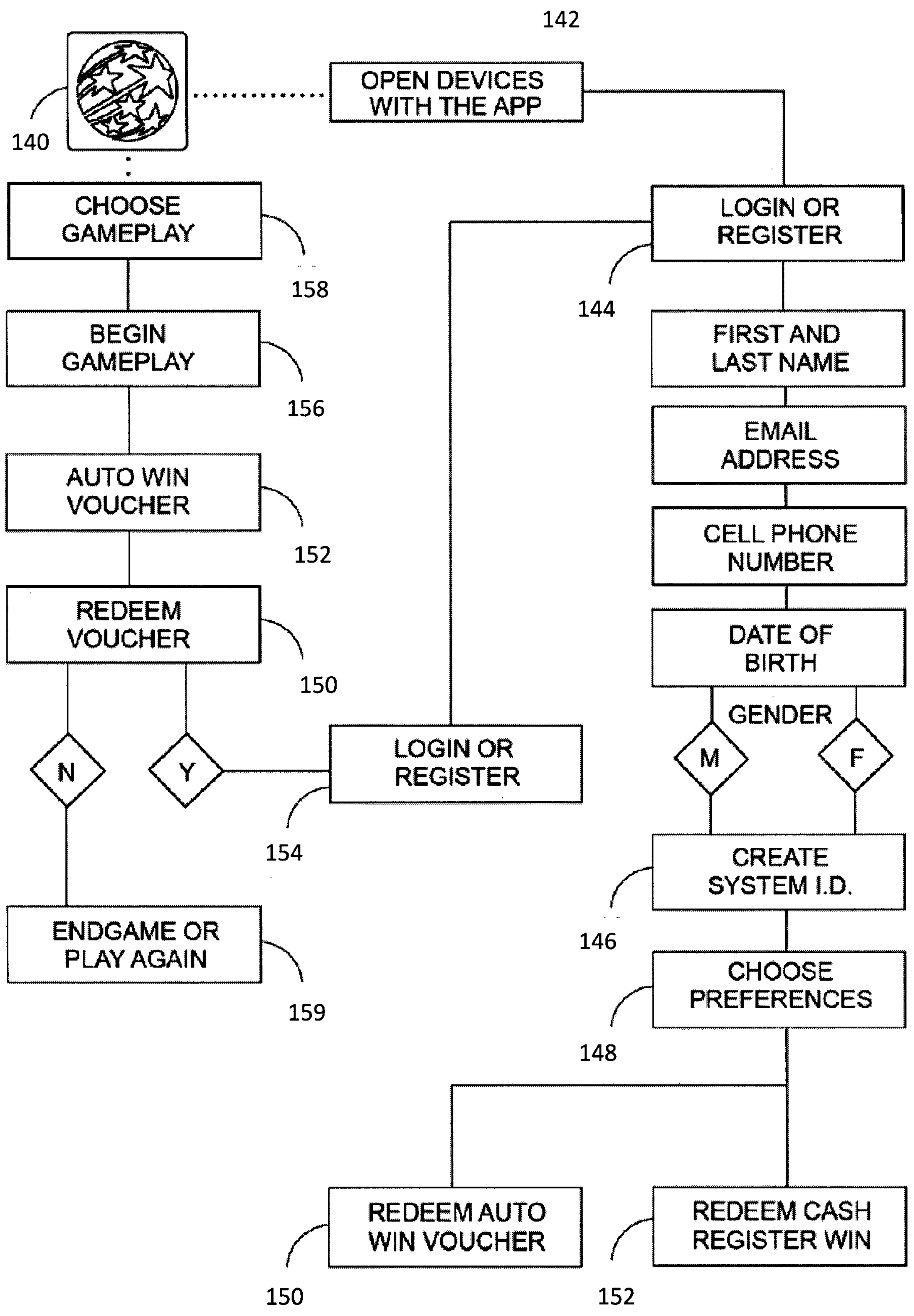
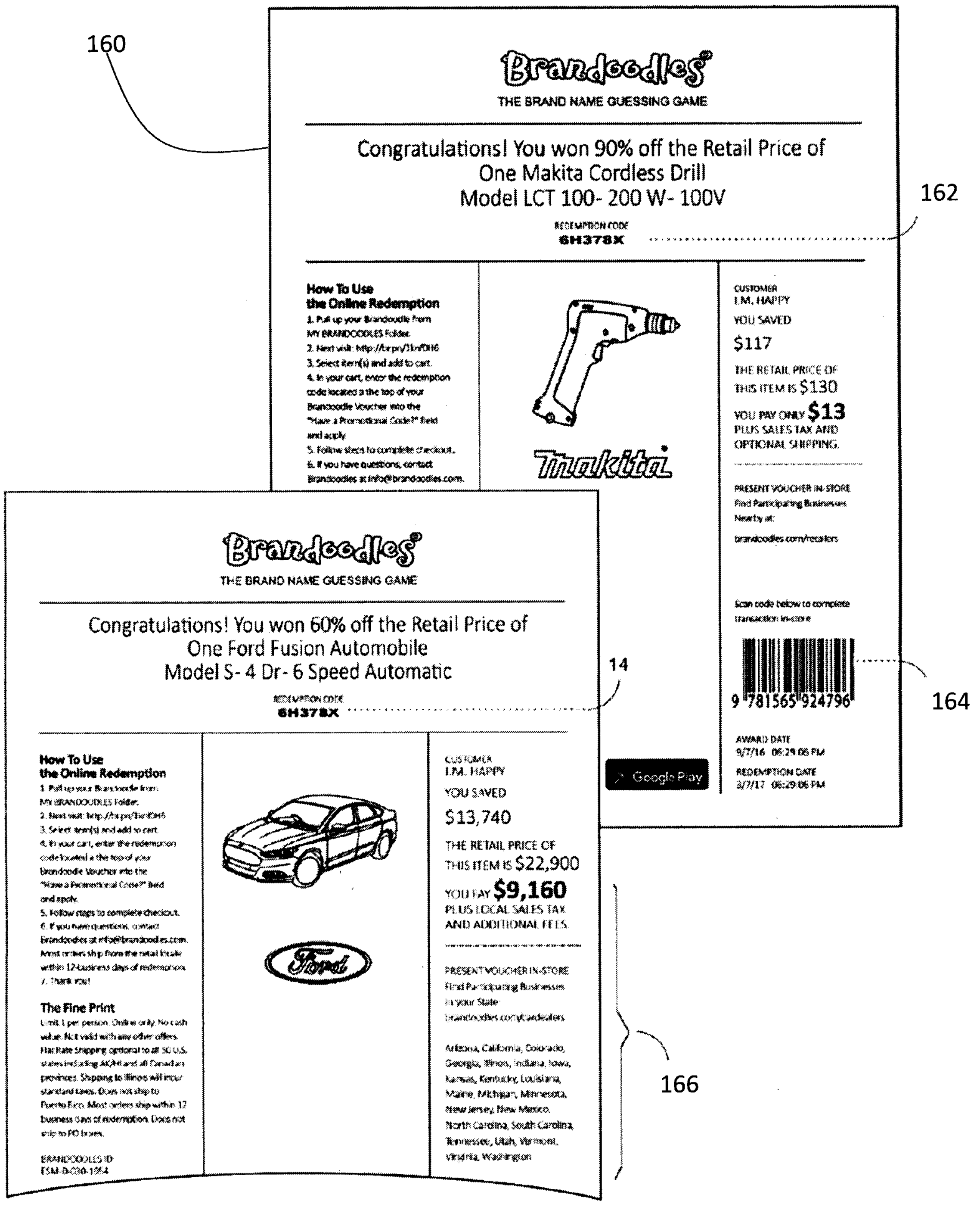

142
OPEN DEVICES WITH THE APP
140
CHOOSE GAMEPLAY
158
BEGIN GAMEPLAY
156
AUTO WIN VOUCHER
152
REDEEM VOUCHER
150
N
Y
LOGIN OR REGISTER
154
ENDGAME OR PLAY AGAIN
159
LOGIN OR REGISTER
144
FIRST AND LAST NAME
EMAIL ADDRESS
CELL PHONE NUMBER
DATE OF BIRTH
GENDER
M
F
CREATE SYSTEM I.D.
146
CHOOSE PREFERENCES
148
REDEEM AUTO WIN VOUCHER
150
REDEEM CASH REGISTER WIN
152

Fig. 6


160
162
164
14
166
Google Play

Brandoodles®
THE BRAND NAME GUESSING GAME

Congratulations! You won 90% off the Retail Price of
One Makita Cordless Drill
Model LCT 100- 200 W- 100V

REDEMPTION CODE
**6H378X**

**How To Use**
**the Online Redemption**
1. Pull up your Brandoodle from MY BRANDOODLES Folder.
2. Next visit: http://bcpn/1kr/DH6
3. Select item(s) and add to cart.
4. In your cart, enter the redemption code located a the top of your Brandoodle Voucher into the "Have a Promotional Code?" field and apply.
5. Follow steps to complete checkout.
6. If you have questions, contact Brandoodles at info@brandoodles.com.

makita

CUSTOMER
I.M. HAPPY
YOU SAVED
$117
THE RETAIL PRICE OF THIS ITEM IS $130
YOU PAY ONLY **$13**
PLUS SALES TAX AND OPTIONAL SHIPPING.

PRESENT VOUCHER IN-STORE
Find Participating Businesses Nearby at:
brandoodles.com/retailers

Scan code below to complete transaction in-store

9 781565 924796

AWARD DATE
9/7/16 06:29:06 PM
REDEMPTION DATE
3/7/17 06:29:06 PM

Brandoodles®
THE BRAND NAME GUESSING GAME

Congratulations! You won 60% off the Retail Price of
One Ford Fusion Automobile
Model S- 4 Dr- 6 Speed Automatic

REDEMPTION CODE
**6H378X**

**How To Use**
**the Online Redemption**
1. Pull up your Brandoodle from MY BRANDOODLES Folder.
2. Next visit: http://bcpn/1kr/DH6
3. Select item(s) and add to cart.
4. In your cart, enter the redemption code located a the top of your Brandoodle Voucher into the "Have a Promotional Code?" field and apply.
5. Follow steps to complete checkout.
6. If you have questions, contact Brandoodles at info@brandoodles.com. Most orders ship from the retail locale within 12-business days of redemption.
7. Thank you!

**The Fine Print**
Limit 1 per person. Online only. No cash value. Not valid with any other offers. Flat Rate Shipping optional to all 50 U.S. states including AK/HI and all Canadian provinces. Shipping to Illinois will incur standard taxes. Does not ship to Puerto Rico. Most orders ship within 12 business days of redemption. Does not ship to PO boxes.

BRANDOODLES ID
ESM-D-030-1954

Ford

CUSTOMER
I.M. HAPPY
YOU SAVED
$13,740
THE RETAIL PRICE OF THIS ITEM IS $22,900
YOU PAY **$9,160**
PLUS LOCAL SALES TAX AND ADDITIONAL FEES.

PRESENT VOUCHER IN-STORE
Find Participating Businesses in your State
brandoodles.com/cardealers

Arizona, California, Colorado, Georgia, Illinois, Indiana, Iowa, Kansas, Kentucky, Louisiana, Maine, Michigan, Minnesota, New Jersey, New Mexico, North Carolina, South Carolina, Tennessee, Utah, Vermont, Virginia, Washington

Fig. 7

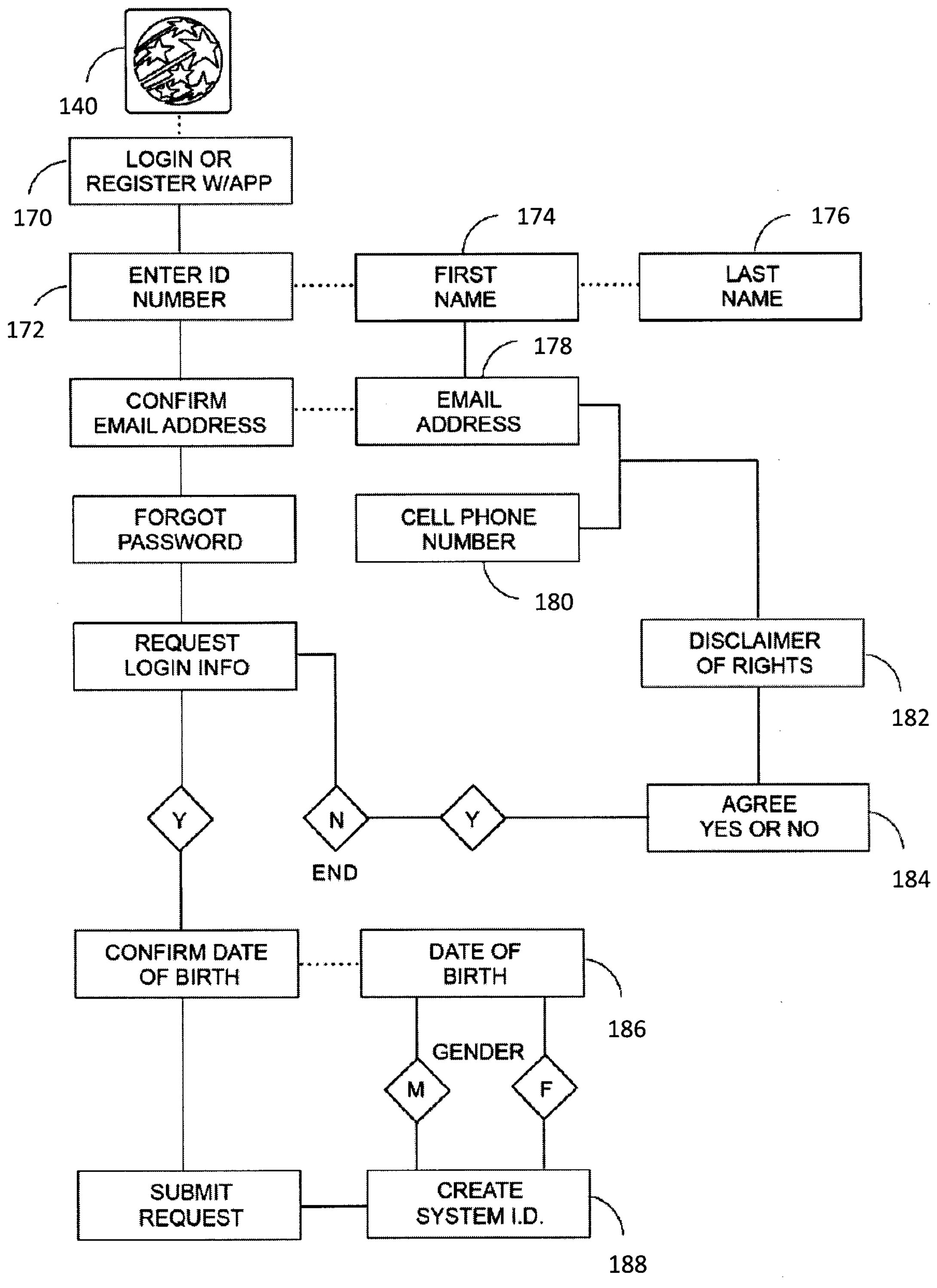
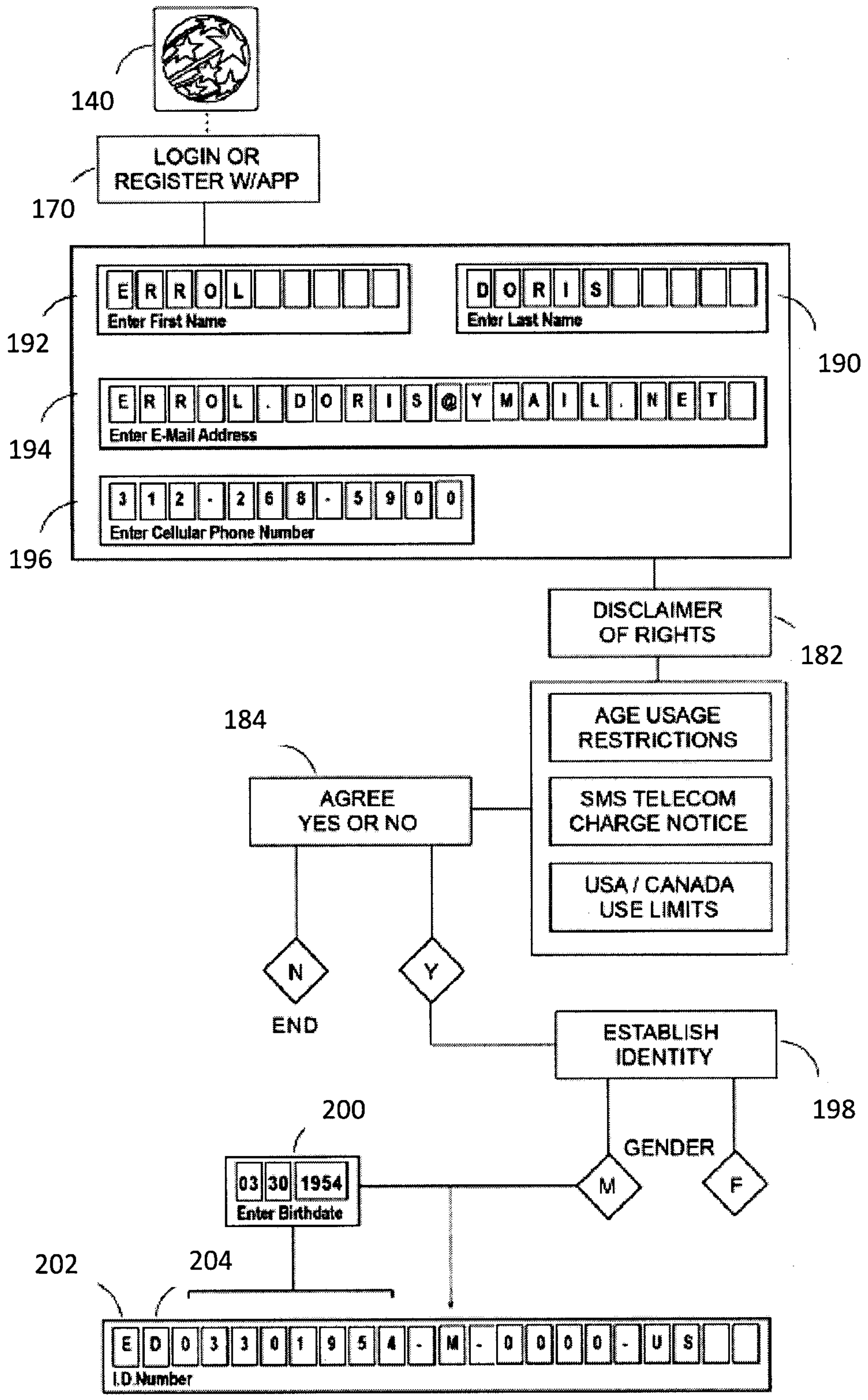

140
LOGIN OR REGISTER W/APP
170
ENTER ID NUMBER
172
FIRST NAME
174
LAST NAME
176
CONFIRM EMAIL ADDRESS
EMAIL ADDRESS
178
FORGOT PASSWORD
CELL PHONE NUMBER
180
REQUEST LOGIN INFO
DISCLAIMER OF RIGHTS
182
Y
N
END
Y
AGREE YES OR NO
184
CONFIRM DATE OF BIRTH
DATE OF BIRTH
186
GENDER
M
F
SUBMIT REQUEST
CREATE SYSTEM I.D.
188

Fig. 8



140
LOGIN OR REGISTER W/APP
170
E R R O L
Enter First Name
D O R I S
Enter Last Name
192
190
E R R O L . D O R I S @ Y M A I L . N E T
Enter E-Mail Address
194
3 1 2 - 2 6 8 - 5 9 0 0
Enter Cellular Phone Number
196
DISCLAIMER OF RIGHTS
182
184
AGE USAGE RESTRICTIONS
AGREE YES OR NO
SMS TELECOM CHARGE NOTICE
USA / CANADA USE LIMITS
N
Y
END
ESTABLISH IDENTITY
198
200
GENDER
03 30 1954
Enter Birthdate
M
F
202
204
E D 0 3 3 0 1 9 5 4 - M - 0 0 0 0 - U S
I.D Number

Fig. 9

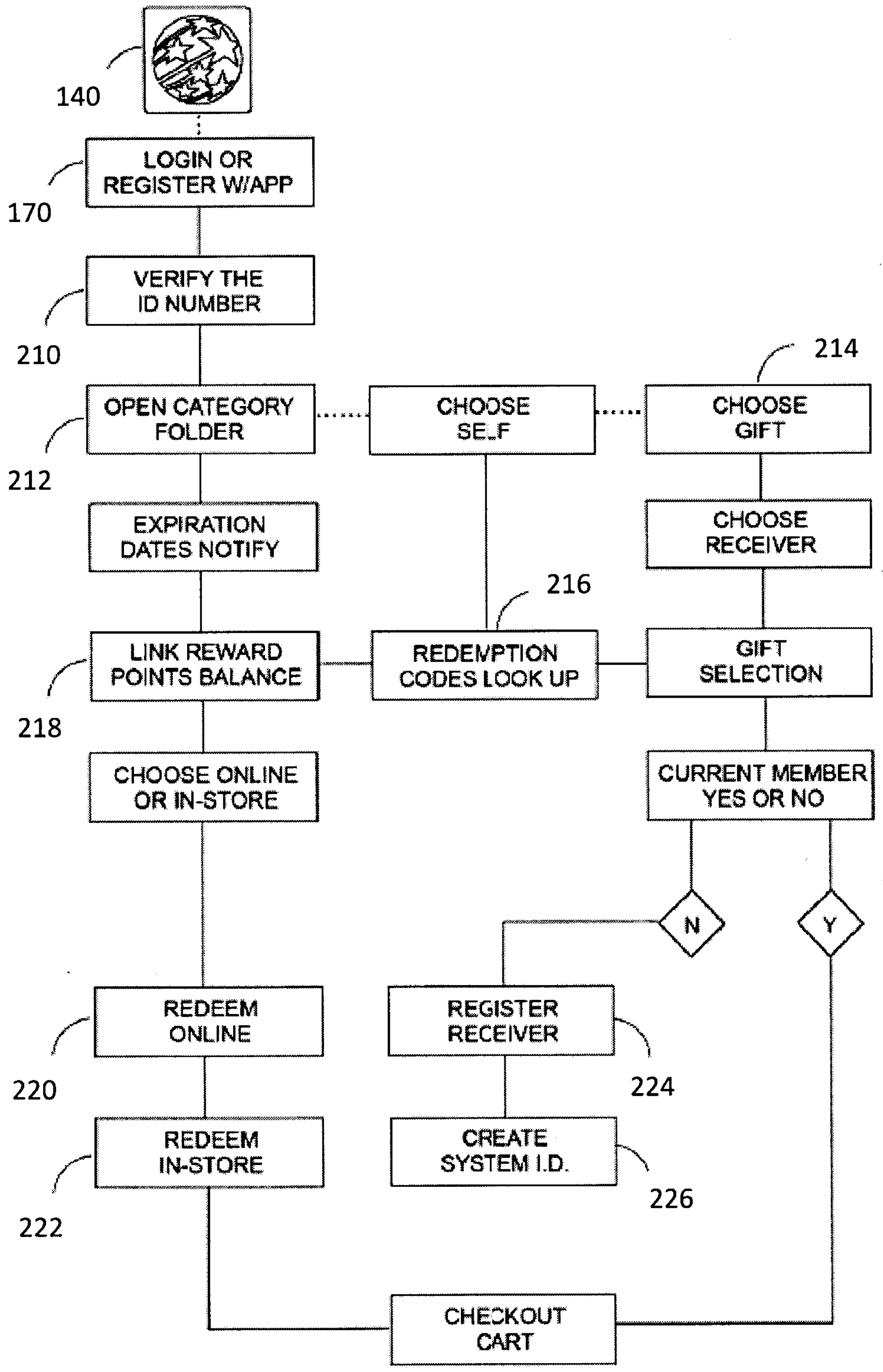
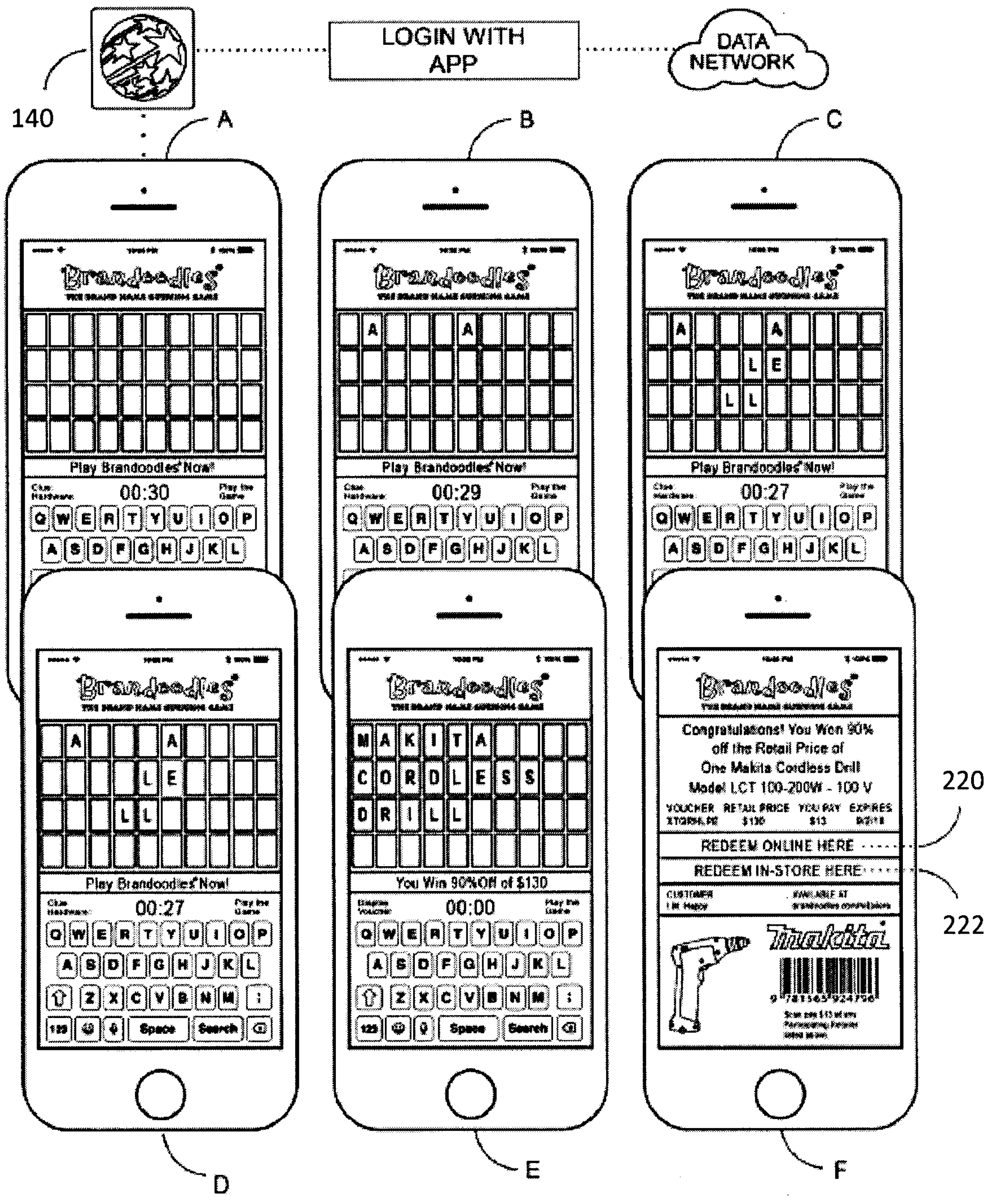

140
LOGIN OR REGISTER W/APP
170
VERIFY THE ID NUMBER
210
OPEN CATEGORY FOLDER
212
CHOOSE SELF
CHOOSE GIFT
214
EXPIRATION DATES NOTIFY
CHOOSE RECEIVER
216
LINK REWARD POINTS BALANCE
218
REDEMPTION CODES LOOK UP
GIFT SELECTION
CHOOSE ONLINE OR IN-STORE
CURRENT MEMBER YES OR NO
N
Y
REDEEM ONLINE
220
REGISTER RECEIVER
224
REDEEM IN-STORE
222
CREATE SYSTEM I.D.
226
CHECKOUT CART

Fig. 10



LOGIN WITH APP
DATA NETWORK
140
A
B
C
D
E
F
220
222
Brandoodles
Play Brandoodles Now!
00:30
00:29
00:27
00:00
You Win 90%Off of $130
Congratulations! You Won 90% off the Retail Price of One Makita Cordless Drill Model LCT 100-200W - 100 V
REDEEM ONLINE HERE
REDEEM IN-STORE HERE
makita

Fig. 11

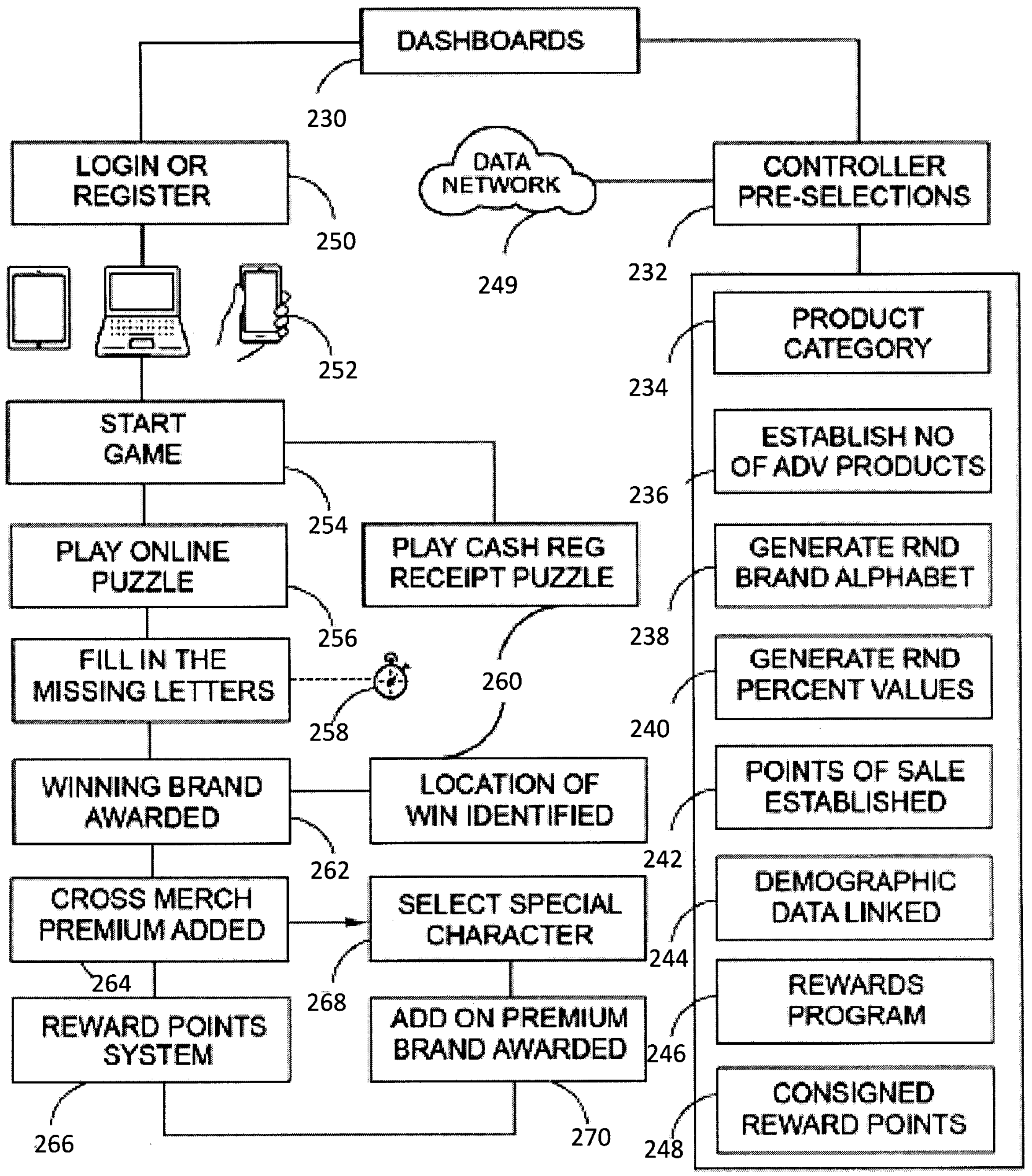

DASHBOARDS
230
LOGIN OR REGISTER
250
DATA NETWORK
249
CONTROLLER PRE-SELECTIONS
232
252
PRODUCT CATEGORY
234
START GAME
254
ESTABLISH NO OF ADV PRODUCTS
236
PLAY ONLINE PUZZLE
256
PLAY CASH REG RECEIPT PUZZLE
GENERATE RND BRAND ALPHABET
238
FILL IN THE MISSING LETTERS
258
260
GENERATE RND PERCENT VALUES
240
WINNING BRAND AWARDED
262
LOCATION OF WIN IDENTIFIED
POINTS OF SALE ESTABLISHED
242
CROSS MERCH PREMIUM ADDED
264
SELECT SPECIAL CHARACTER
268
DEMOGRAPHIC DATA LINKED
244
REWARD POINTS SYSTEM
266
ADD ON PREMIUM BRAND AWARDED
270
REWARDS PROGRAM
246
CONSIGNED REWARD POINTS
248

Fig. 12

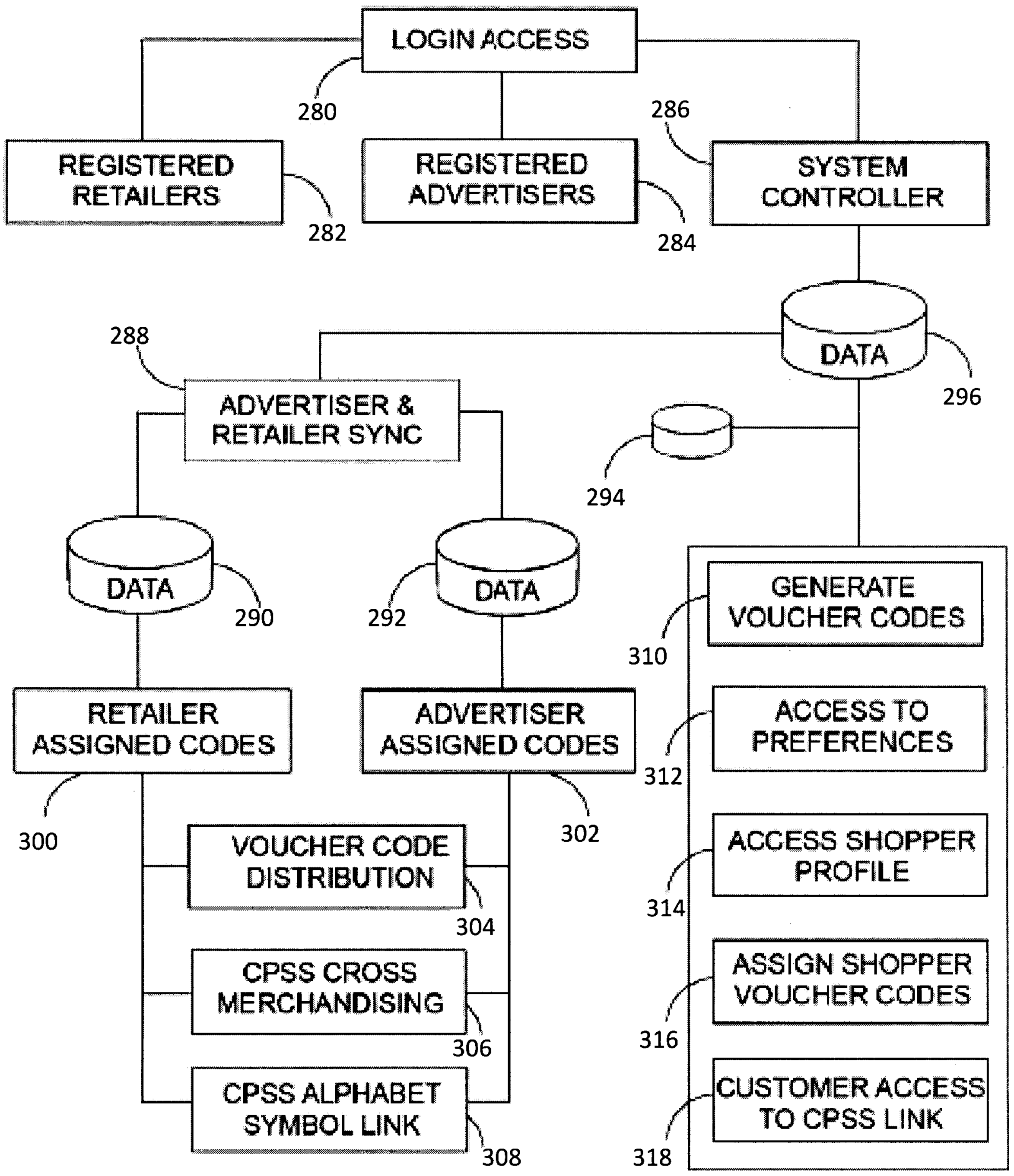

LOGIN ACCESS
280
REGISTERED RETAILERS
282
REGISTERED ADVERTISERS
284
286
SYSTEM CONTROLLER
DATA
296
288
ADVERTISER & RETAILER SYNC
294
DATA
290
DATA
292
GENERATE VOUCHER CODES
310
ACCESS TO PREFERENCES
312
ACCESS SHOPPER PROFILE
314
ASSIGN SHOPPER VOUCHER CODES
316
CUSTOMER ACCESS TO CPSS LINK
318
RETAILER ASSIGNED CODES
300
ADVERTISER ASSIGNED CODES
302
VOUCHER CODE DISTRIBUTION
304
CPSS CROSS MERCHANDISING
306
CPSS ALPHABET SYMBOL LINK
308

Fig. 13

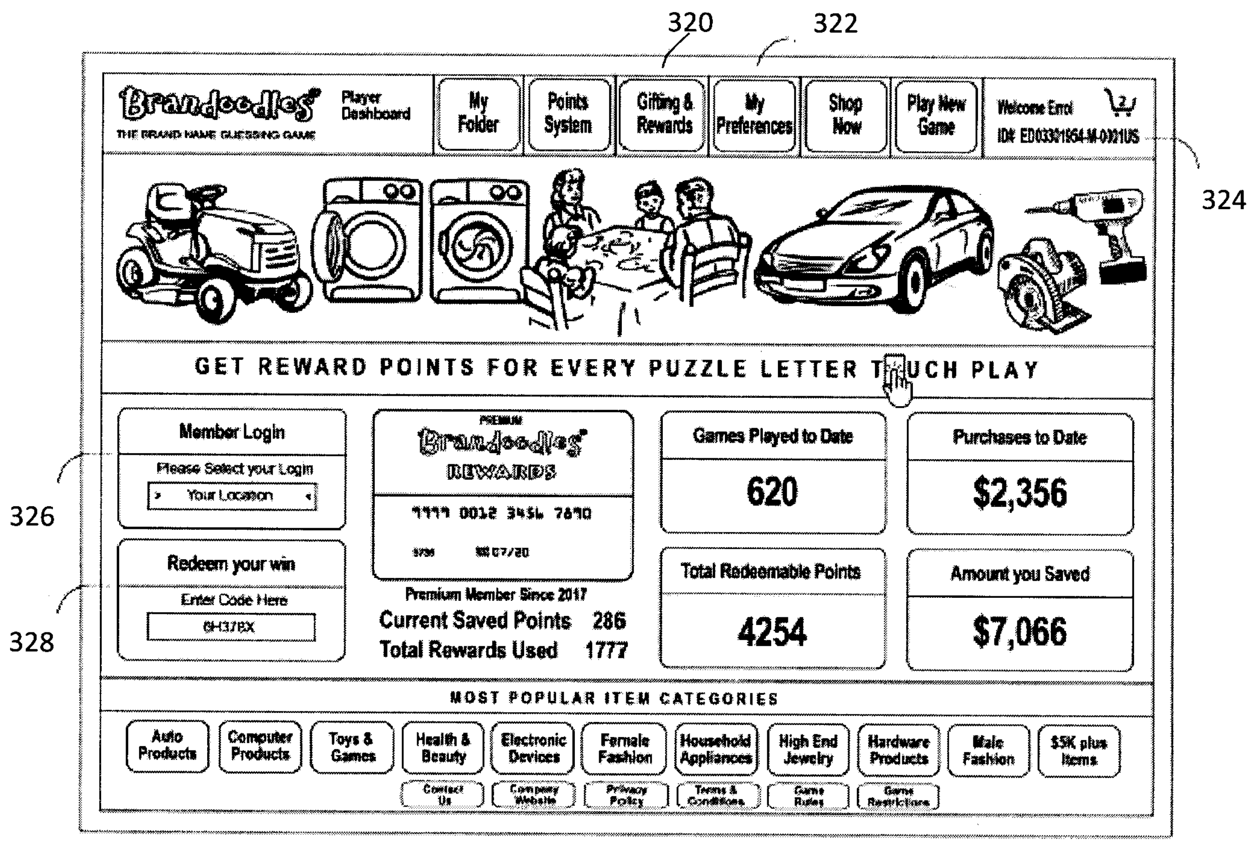
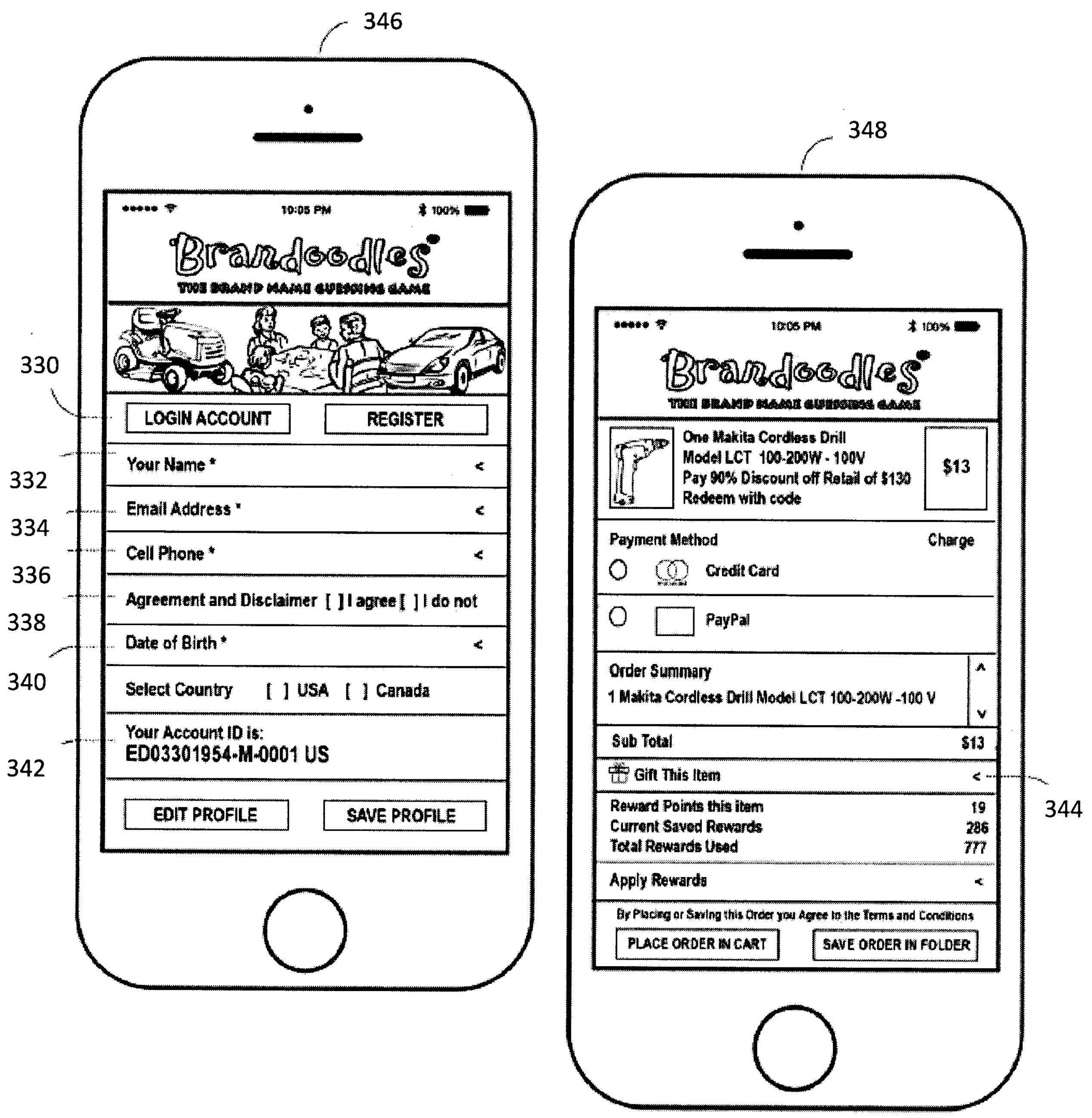

320
322
324
326
328
Brandoodles
THE BRAND NAME GUESSING GAME
Player Dashboard
My Folder
Points System
Gifting & Rewards
My Preferences
Shop Now
Play New Game
Welcome Errol
ID# ED03301964-M-0001US
GET REWARD POINTS FOR EVERY PUZZLE LETTER TOUCH PLAY
Member Login
Please Select your Login
Your Location
Redeem your win
Enter Code Here
6H378X
PREMIUM
Brandoodles
REWARDS
7777 0012 3456 7890
07/20
Premium Member Since 2017
Current Saved Points 286
Total Rewards Used 1777
Games Played to Date
620
Purchases to Date
$2,356
Total Redeemable Points
4254
Amount you Saved
$7,066
MOST POPULAR ITEM CATEGORIES
Auto Products
Computer Products
Toys & Games
Health & Beauty
Electronic Devices
Female Fashion
Household Appliances
High End Jewelry
Hardware Products
Male Fashion
$5K plus Items
Contact Us
Company Website
Privacy Policy
Terms & Conditions
Game Rules
Game Restrictions

Fig. 14



10:05 PM 100%

Brandoodles

THE BRAND NAME GUESSING GAME

LOGIN ACCOUNT | REGISTER

Your Name * <

Email Address * <

Cell Phone * <

Agreement and Disclaimer [ ] I agree [ ] I do not

Date of Birth * <

Select Country [ ] USA [ ] Canada

Your Account ID is:
ED03301954-M-0001 US

EDIT PROFILE | SAVE PROFILE

10:05 PM 100%

Brandoodles

THE BRAND NAME GUESSING GAME

One Makita Cordless Drill
Model LCT 100-200W - 100V
Pay 90% Discount off Retail of $130
Redeem with code

$13

Payment Method | Charge

Credit Card

PayPal

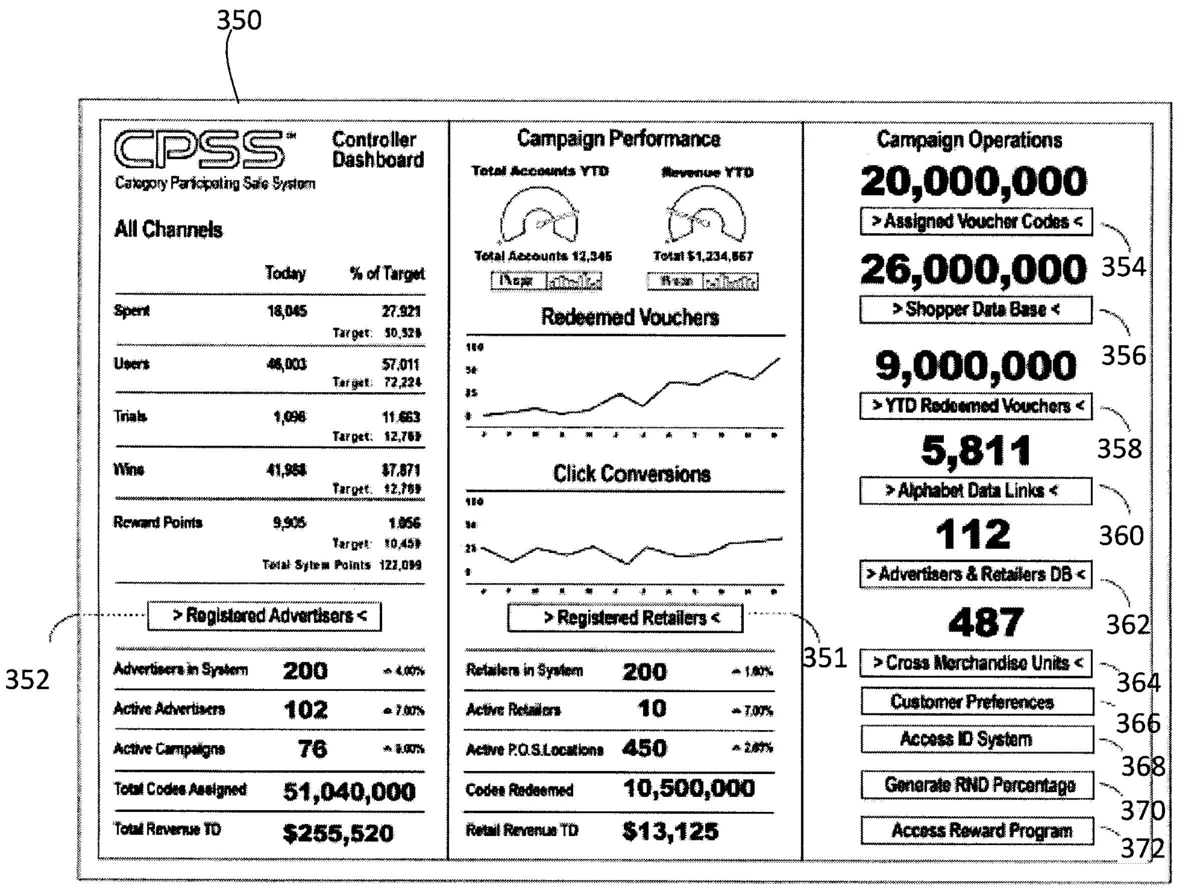
Order Summary
1 Makita Cordless Drill Model LCT 100-200W -100 V

| | |
|---|---|
| Sub Total | $13 |
| Gift This Item | < |
| Reward Points this item | 19 |
| Current Saved Rewards | 286 |
| Total Rewards Used | 777 |
| Apply Rewards | < |

By Placing or Saving this Order you Agree to the Terms and Conditions

PLACE ORDER IN CART | SAVE ORDER IN FOLDER

Fig. 15



CPSS℠ Controller Dashboard
Category Participating Sale System

All Channels

| | Today | % of Target |
|---|---|---|
| Spent | 18,045 | 27.921 |
| | | Target: 50,328 |
| Users | 46,003 | 57,011 |
| | | Target: 72,224 |
| Trials | 1,096 | 11.663 |
| | | Target: 12,769 |
| Wins | 41,958 | 87,871 |
| | | Target: 12,769 |
| Reward Points | 9,905 | 1.056 |
| | | Target: 10,459 |
| | Total Sytem Points | 122,099 |

> Registered Advertisers <

| | | |
|---|---|---|
| Advertisers in System | 200 | ▲ 4.00% |
| Active Advertisers | 102 | ▲ 7.00% |
| Active Campaigns | 76 | ▲ 8.00% |
| Total Codes Assigned | 51,040,000 | |
| Total Revenue TD | $255,520 | |

Campaign Performance

Total Accounts YTD
Total Accounts 12,345

Revenue YTD
Total $1,234,567

Redeemed Vouchers

Click Conversions

> Registered Retailers <

| | | |
|---|---|---|
| Retailers in System | 200 | ▲ 1.80% |
| Active Retailers | 10 | ▲ 7.00% |
| Active P.O.S.Locations | 450 | ▲ 2.69% |
| Codes Redeemed | 10,500,000 | |
| Retail Revenue TD | $13,125 | |

Campaign Operations

20,000,000
> Assigned Voucher Codes <

26,000,000
> Shopper Data Base <

9,000,000
> YTD Redeemed Vouchers <

5,811
> Alphabet Data Links <

112
> Advertisers & Retailers DB <

487
> Cross Merchandise Units <

Customer Preferences

Access ID System

Generate RND Percentage

Access Reward Program

Fig. 16

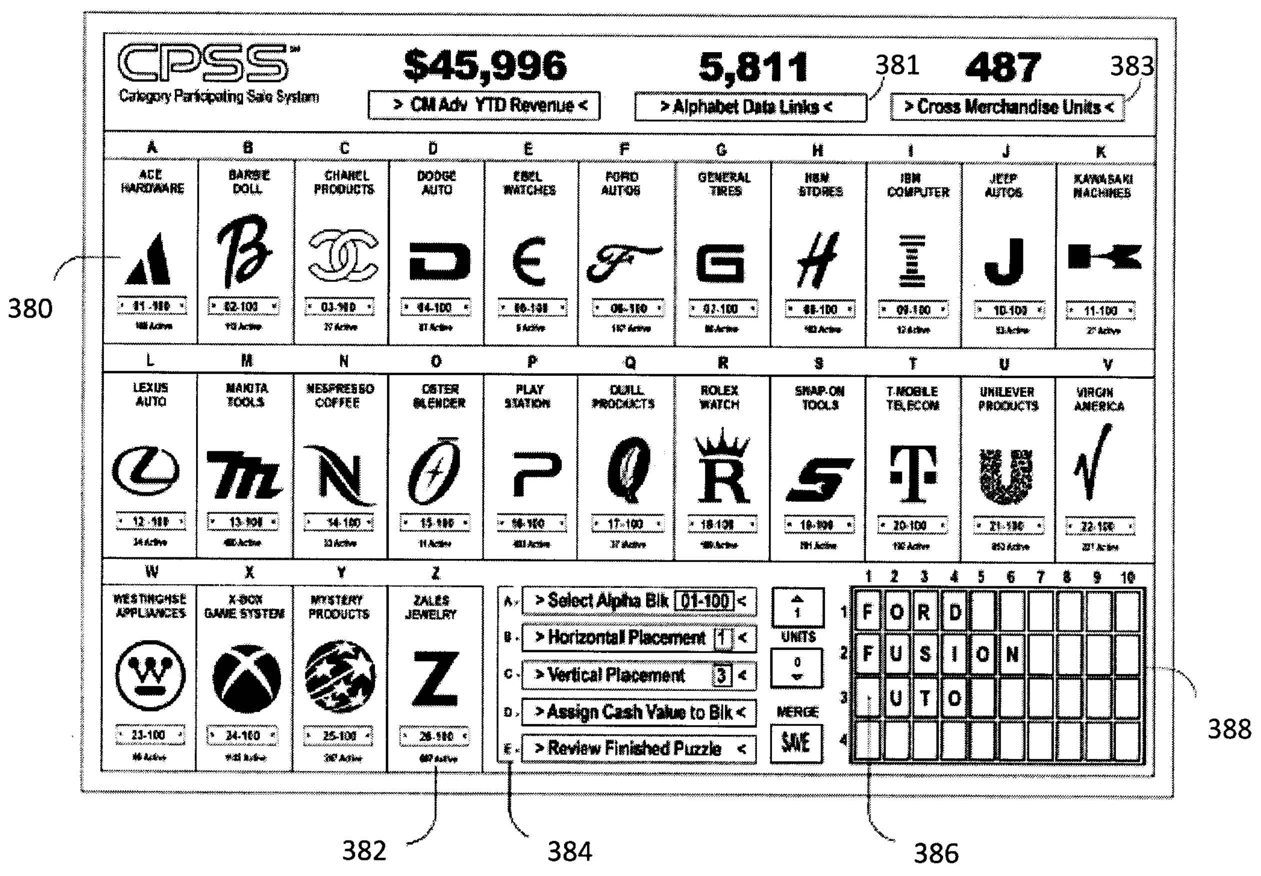

CPSS℠
Category Participating Sale System
$45,996
> CM Adv YTD Revenue <
5,811
> Alphabet Data Links <
487
> Cross Merchandise Units <
381
383
380
A ACE HARDWARE
B BARBIE DOLL
C CHANEL PRODUCTS
D DODGE AUTO
E EBEL WATCHES
F FORD AUTOS
G GENERAL TIRES
H H&M STORES
I IBM COMPUTER
J JEEP AUTOS
K KAWASAKI MACHINES
L LEXUS AUTO
M MAKITA TOOLS
N NESPRESSO COFFEE
O OSTER BLENDER
P PLAY STATION
Q QUILL PRODUCTS
R ROLEX WATCH
S SNAP-ON TOOLS
T T-MOBILE TELECOM
U UNILEVER PRODUCTS
V VIRGIN AMERICA
W WESTINGHSE APPLIANCES
X X-BOX GAME SYSTEM
Y MYSTERY PRODUCTS
Z ZALES JEWELRY
A. > Select Alpha Blk 01-100 <
B. > Horizontal Placement 1 <
C. > Vertical Placement 3 <
D. > Assign Cash Value to Blk <
E. > Review Finished Puzzle <
UNITS
MERGE
SAVE
1 2 3 4 5 6 7 8 9 10
F O R D
F U S I O N
A U T O
382
384
386
388

Fig. 17

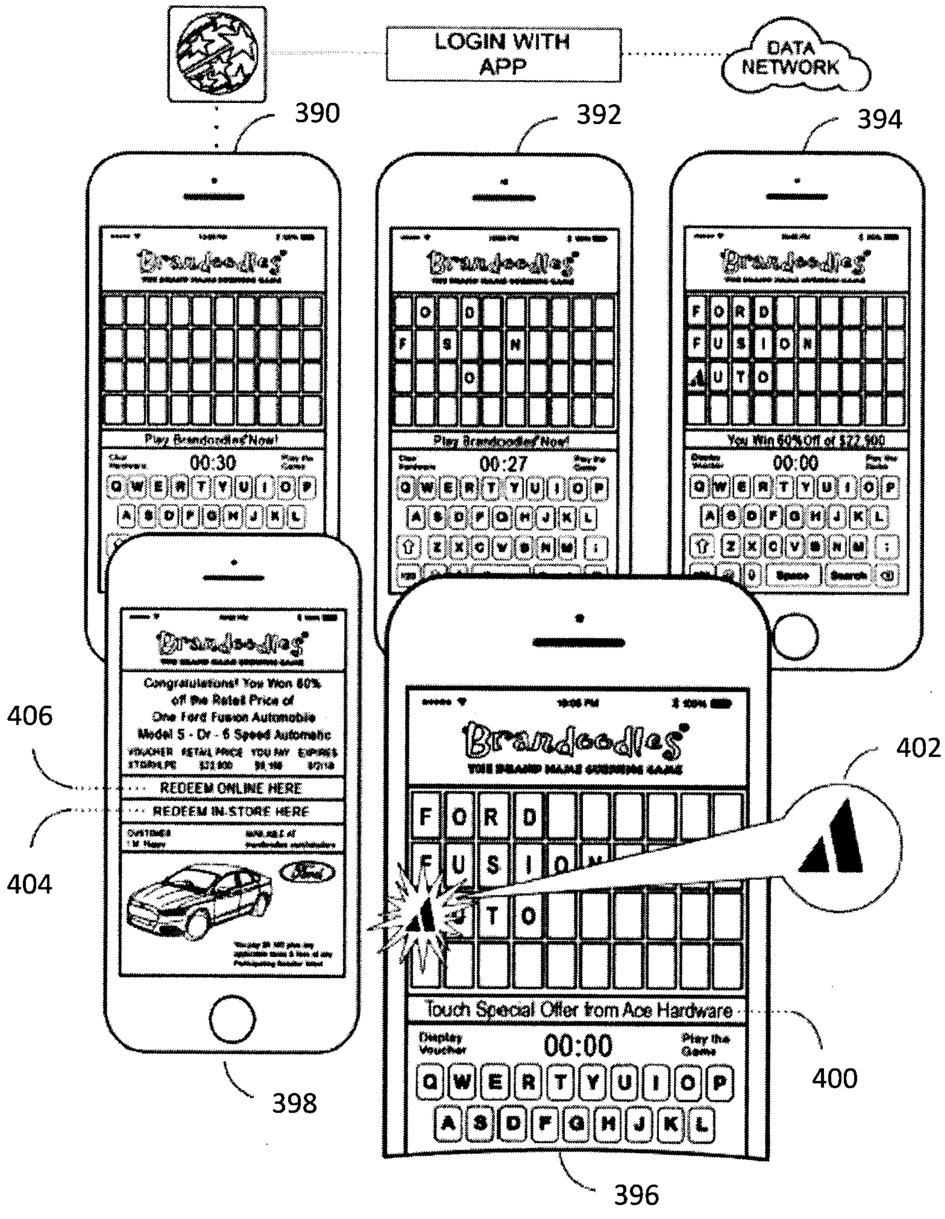

LOGIN WITH APP
DATA NETWORK
390
392
394
396
398
400
402
404
406
Brandoodles
THE BRAND NAME GUESSING GAME
Play Brandoodles Now!
00:30
00:27
00:00
You Win 60% Off of $22,900
Congratulations! You Won 60% off the Retail Price of One Ford Fusion Automobile Model 5 - Dr - 6 Speed Automatic
REDEEM ONLINE HERE
REDEEM IN-STORE HERE
Touch Special Offer from Ace Hardware
Display Voucher
Play the Game
FORD
FUSION
AUTO

Fig. 18

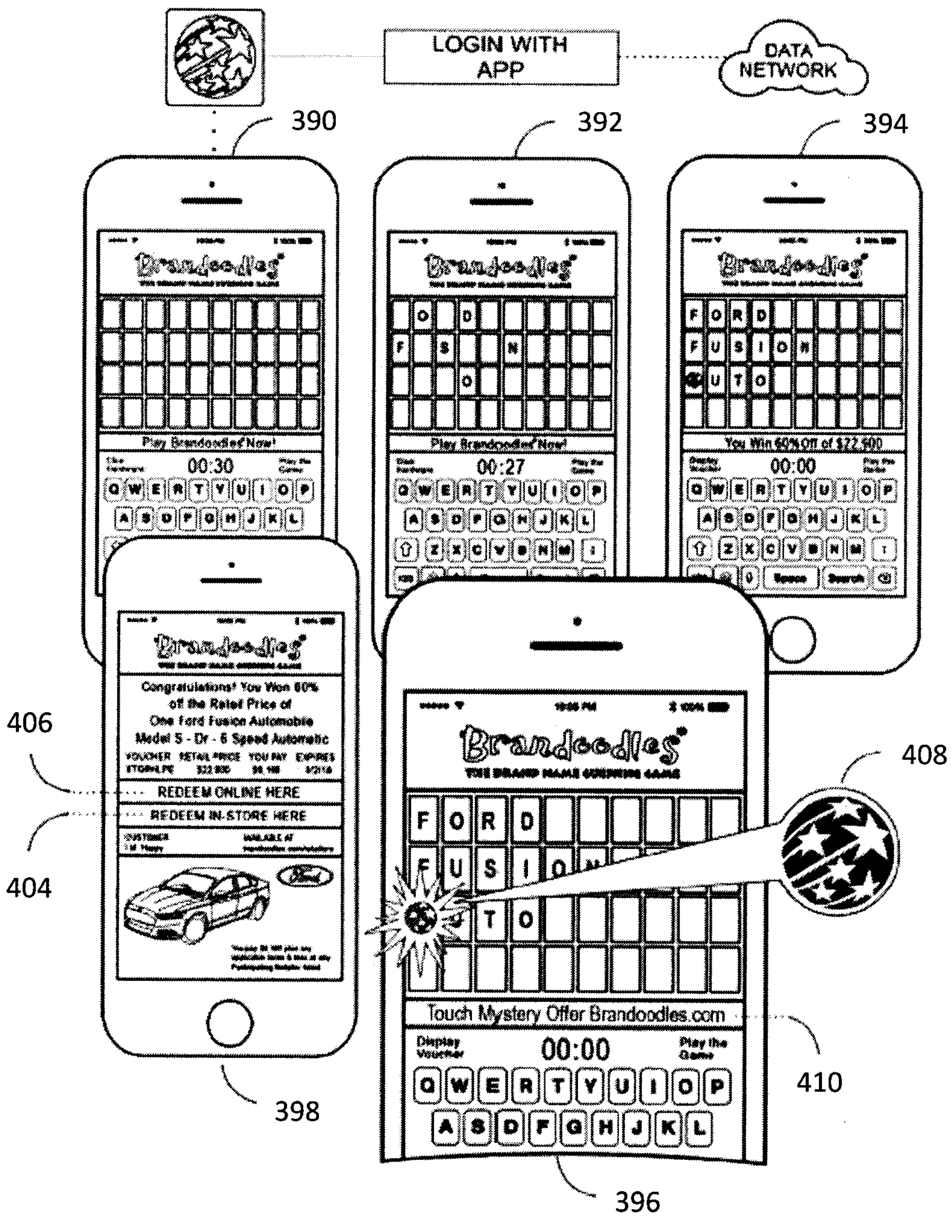
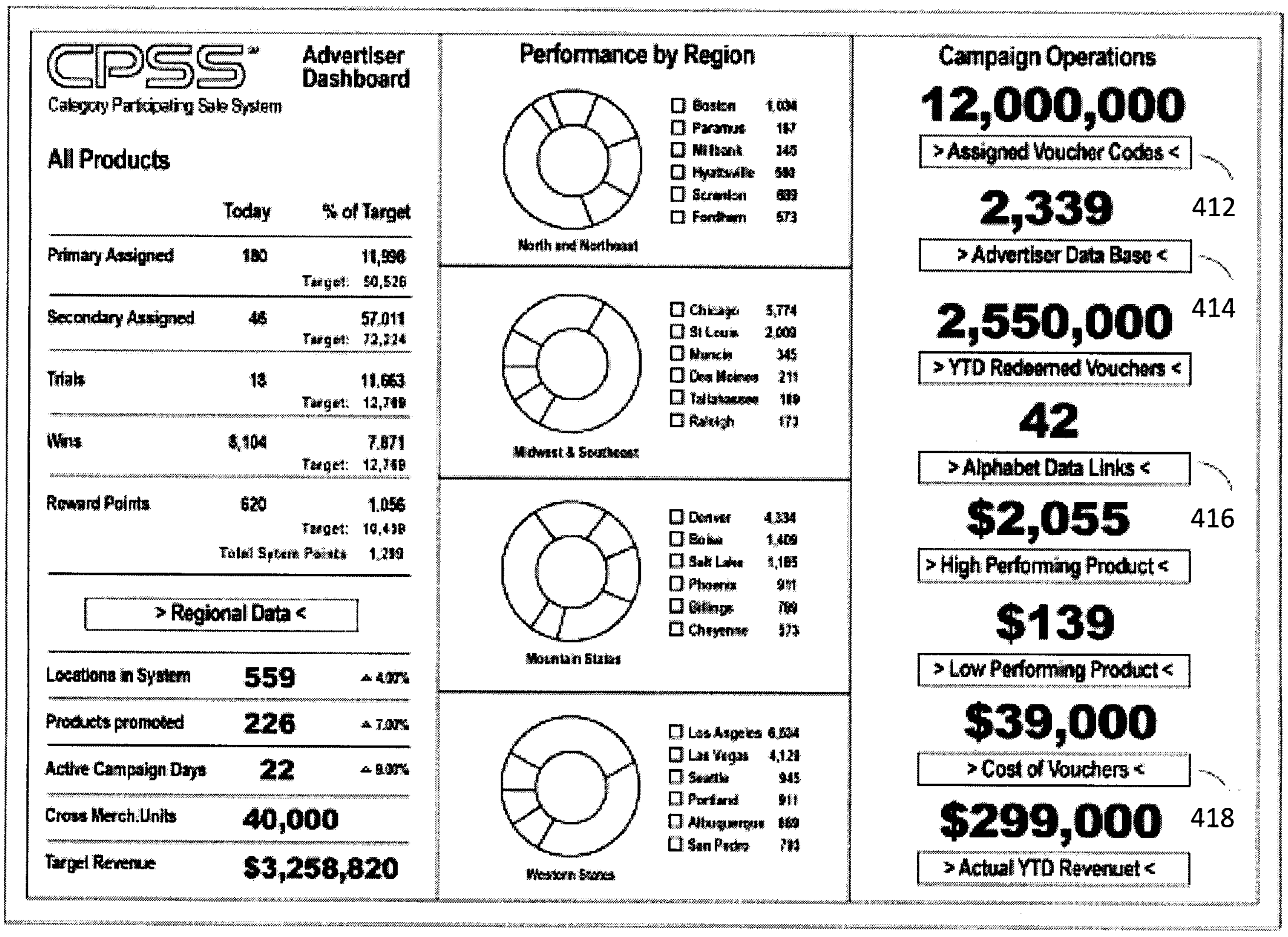

LOGIN WITH APP
DATA NETWORK
390
392
394
Brandoodles
Play Brandoodles Now!
00:30
00:27
You Win 60% Off of $22,500
00:00
406
404
Congratulations! You Won 60% off the Retail Price of One Ford Fusion Automobile Model S - Dr - 6 Speed Automatic
REDEEM ONLINE HERE
REDEEM IN-STORE HERE
398
408
Touch Mystery Offer Brandoodles.com
Display Voucher
Play the Game
410
396

Fig. 19



All Products

| | Today | % of Target |
|---|---|---|
| Primary Assigned | 180 | 11,998 |
| | Target: | 50,526 |
| Secondary Assigned | 46 | 57,011 |
| | Target: | 72,324 |
| Trials | 18 | 11,663 |
| | Target: | 12,749 |
| Wins | 8,104 | 7,871 |
| | Target: | 12,749 |
| Reward Points | 620 | 1,056 |
| | Target: | 10,439 |
| | Total Sytem Points | 1,289 |

> Regional Data <

| | | |
|---|---|---|
| Locations in System | 559 | ▲ 4.00% |
| Products promoted | 226 | ▲ 7.00% |
| Active Campaign Days | 22 | ▲ 9.00% |
| Cross Merch.Units | 40,000 | |
| Target Revenue | $3,258,820 | |

Performance by Region

North and Northeast

| | |
|---|---|
| ☐ Boston | 1,034 |
| ☐ Paramus | 187 |
| ☐ Millbank | 345 |
| ☐ Hyattsville | 580 |
| ☐ Scranton | 639 |
| ☐ Fordham | 573 |

Midwest & Southeast

| | |
|---|---|
| ☐ Chicago | 5,774 |
| ☐ St Louis | 2,009 |
| ☐ Muncie | 345 |
| ☐ Des Moines | 211 |
| ☐ Tallahassee | 189 |
| ☐ Raleigh | 173 |

Mountain States

| | |
|---|---|
| ☐ Denver | 4,334 |
| ☐ Boise | 1,409 |
| ☐ Salt Lake | 1,185 |
| ☐ Phoenix | 911 |
| ☐ Billings | 789 |
| ☐ Cheyenne | 573 |

Western States

| | |
|---|---|
| ☐ Los Angeles | 6,534 |
| ☐ Las Vegas | 4,128 |
| ☐ Seattle | 945 |
| ☐ Portland | 911 |
| ☐ Albuquerque | 869 |
| ☐ San Pedro | 793 |

Campaign Operations

12,000,000

> Assigned Voucher Codes <

2,339

> Advertiser Data Base <

2,550,000

> YTD Redeemed Vouchers <

42

> Alphabet Data Links <

$2,055

> High Performing Product <

$139

> Low Performing Product <

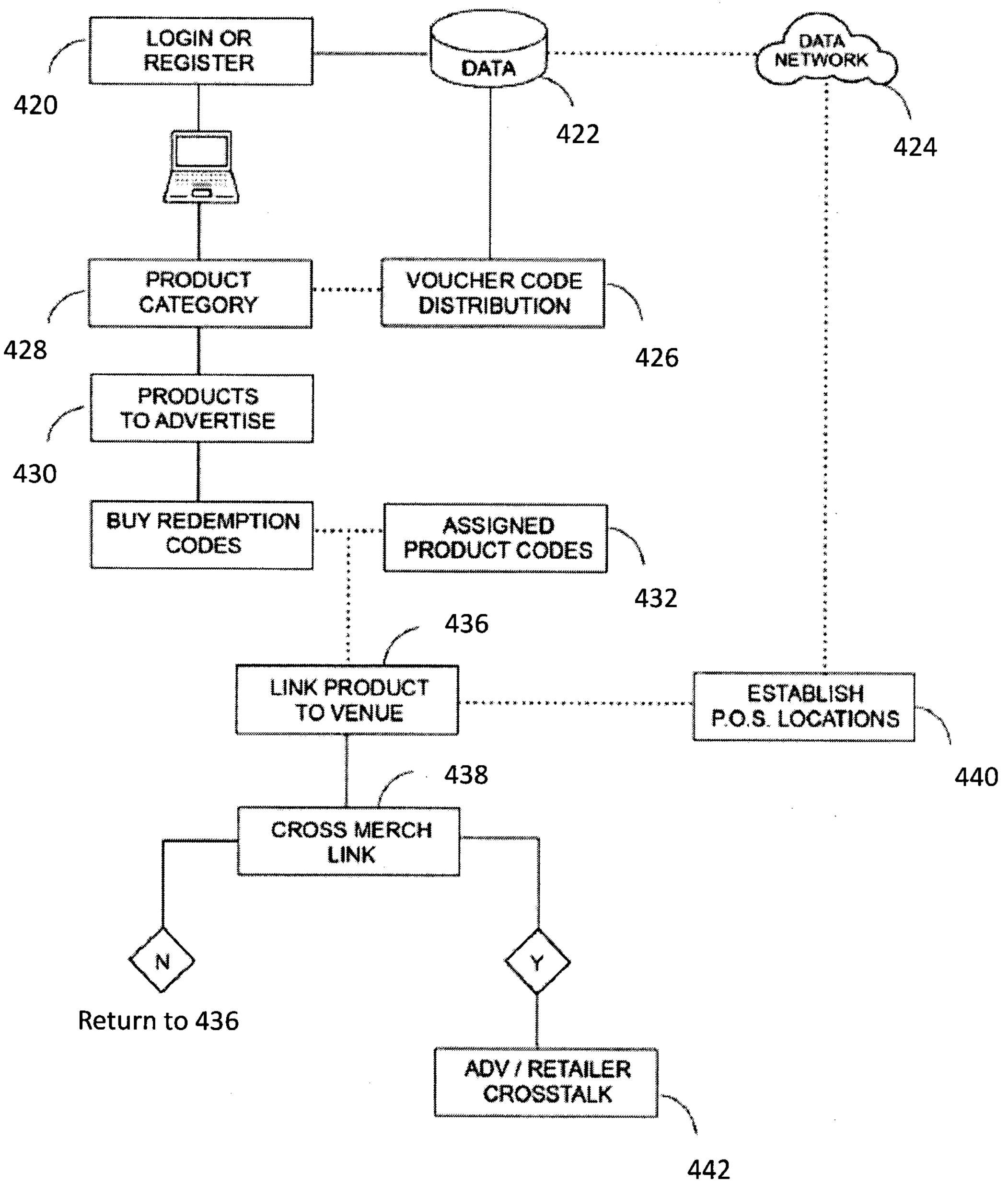
$39,000

> Cost of Vouchers <

$299,000

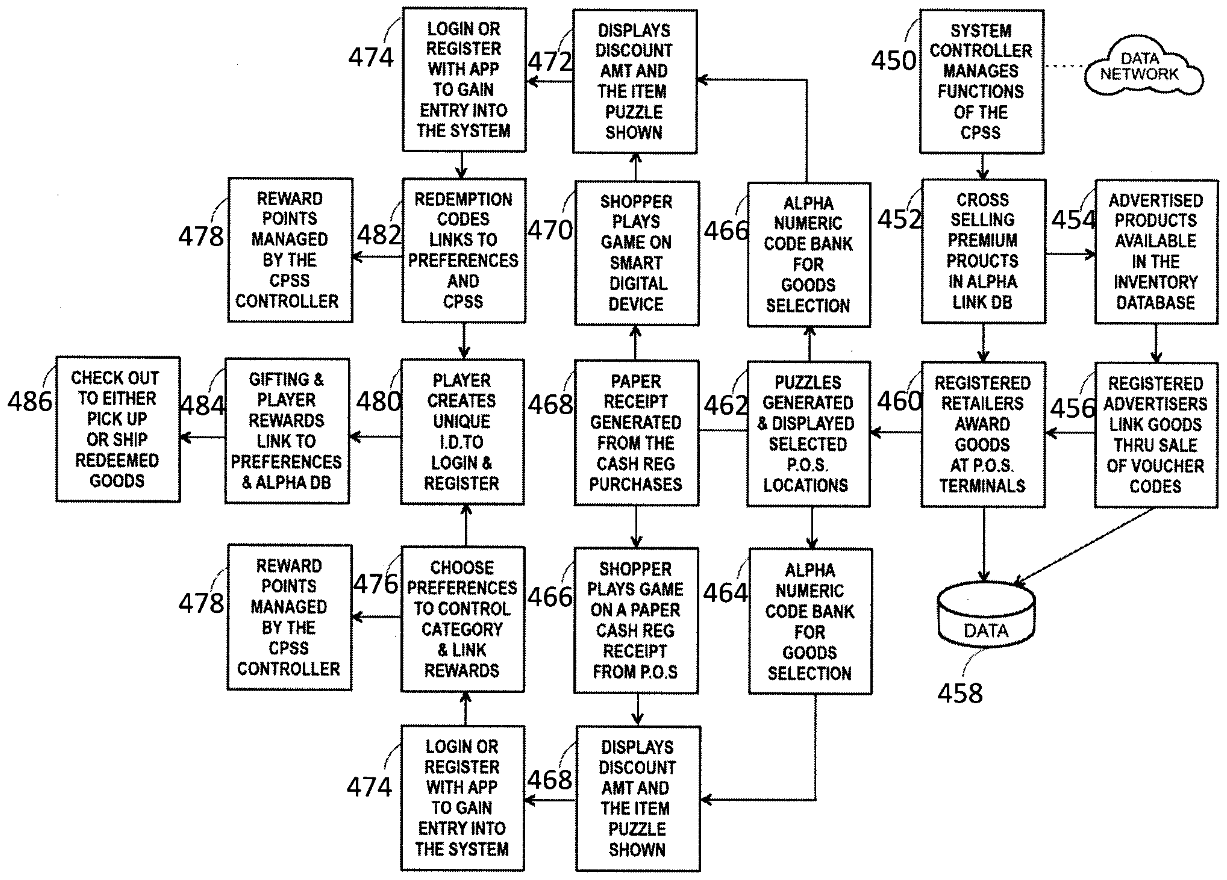
> Actual YTD Revenuet <

Fig. 20



LOGIN OR REGISTER
420
DATA
422
DATA NETWORK
424
PRODUCT CATEGORY
428
VOUCHER CODE DISTRIBUTION
426
PRODUCTS TO ADVERTISE
430
BUY REDEMPTION CODES
ASSIGNED PRODUCT CODES
432
436
LINK PRODUCT TO VENUE
ESTABLISH P.O.S. LOCATIONS
440
438
CROSS MERCH LINK
N
Return to 436
Y
ADV / RETAILER CROSSTALK
442

Fig. 21



450 SYSTEM CONTROLLER MANAGES FUNCTIONS OF THE CPSS
DATA NETWORK
452 CROSS SELLING PREMIUM PROUCTS IN ALPHA LINK DB
454 ADVERTISED PRODUCTS AVAILABLE IN THE INVENTORY DATABASE
456 REGISTERED ADVERTISERS LINK GOODS THRU SALE OF VOUCHER CODES
460 REGISTERED RETAILERS AWARD GOODS AT P.O.S. TERMINALS
DATA
458
462 PUZZLES GENERATED & DISPLAYED SELECTED P.O.S. LOCATIONS
466 ALPHA NUMERIC CODE BANK FOR GOODS SELECTION
464 ALPHA NUMERIC CODE BANK FOR GOODS SELECTION
472 DISPLAYS DISCOUNT AMT AND THE ITEM PUZZLE SHOWN
470 SHOPPER PLAYS GAME ON SMART DIGITAL DEVICE
468 PAPER RECEIPT GENERATED FROM THE CASH REG PURCHASES
466 SHOPPER PLAYS GAME ON A PAPER CASH REG RECEIPT FROM P.O.S
468 DISPLAYS DISCOUNT AMT AND THE ITEM PUZZLE SHOWN
474 LOGIN OR REGISTER WITH APP TO GAIN ENTRY INTO THE SYSTEM
482 REDEMPTION CODES LINKS TO PREFERENCES AND CPSS
480 PLAYER CREATES UNIQUE I.D.TO LOGIN & REGISTER
476 CHOOSE PREFERENCES TO CONTROL CATEGORY & LINK REWARDS
474 LOGIN OR REGISTER WITH APP TO GAIN ENTRY INTO THE SYSTEM
478 REWARD POINTS MANAGED BY THE CPSS CONTROLLER
484 GIFTING & PLAYER REWARDS LINK TO PREFERENCES & ALPHA DB
486 CHECK OUT TO EITHER PICK UP OR SHIP REDEEMED GOODS
478 REWARD POINTS MANAGED BY THE CPSS CONTROLLER

Fig. 22

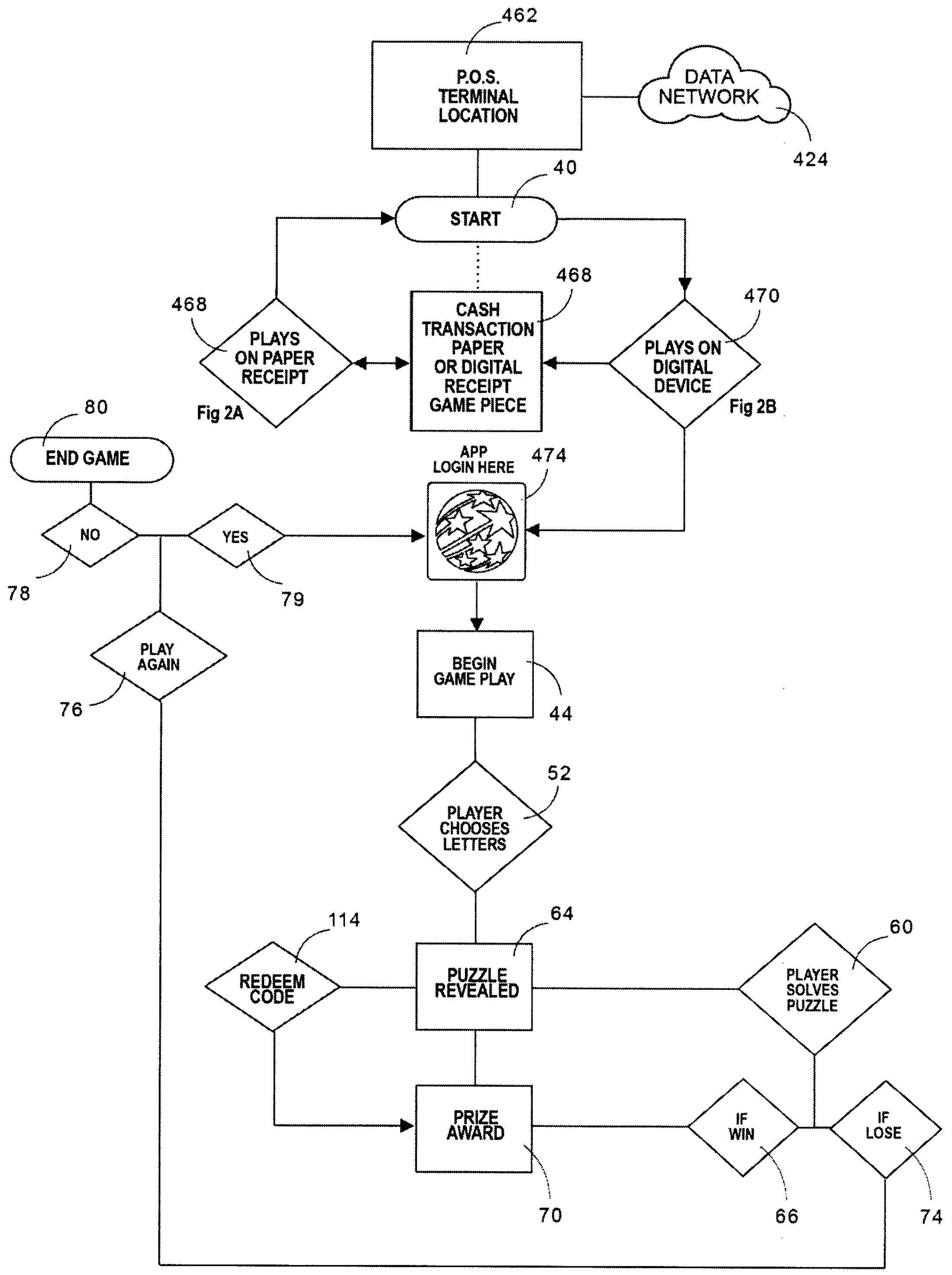

462
P.O.S. TERMINAL LOCATION
DATA NETWORK
424
40
START
468
CASH TRANSACTION PAPER OR DIGITAL RECEIPT GAME PIECE
468
PLAYS ON PAPER RECEIPT
Fig 2A
470
PLAYS ON DIGITAL DEVICE
Fig 2B
80
END GAME
APP LOGIN HERE
474
NO
78
YES
79
PLAY AGAIN
76
BEGIN GAME PLAY
44
52
PLAYER CHOOSES LETTERS
114
64
60
REDEEM CODE
PUZZLE REVEALED
PLAYER SOLVES PUZZLE
PRIZE AWARD
IF WIN
IF LOSE
70
66
74

Fig. 23