data entered. A menu also has a numeric code for each of the storage areas and a special code including a security code for certain menu items, process codes of forms within the menu, or a pointer to the process code. A pointer may also be provided in the menu for processes to be performed off-line (i.e., in an associated database server **28**).

If a process (P_{xy}) is selected at step **104**, the database server **28** is notified that something is requested from its database **30** or that some processing of data is requested. For example, the data transaction assembler **18** may send a user "?" inquiry to the database server **28** so that options may be returned to the data transaction assembler **18** for presentation to the user for selection. The process triggers an external process of database server **30** with a parameter stream, and control is either returned to the data transaction assembler **18** or control is held up until the process is complete, in which case a message is sent back to the data transaction assembler **18**. This message can be a report, selected data, a value resulting from a calculation, and the like. Processing such as checking detectors and the like may also be performed locally by data transaction assembler **18**.

Once the desired form is selected for the user's application, the form is processed at step **110** in accordance with the steps outlined in FIGS. **8–10**. As an entry is made in each field, it is automatically stored within the input buffer area of the transaction buffer **97** at its assigned location and in the dictated format. At any time, the entire form may be exited with automatic return to the menu which called it or the form can be cleared for data reentry. Once the form has been processed and transmitted to the appropriate database server(s) **28**, the database server connection is terminated and the user is presented at step **112** with the last menu from which the user made his or her selection. Alternatively, the executive menu will be called up as a default menu.

If the user indicates at step **114** that he or she wishes to continue to complete a new form, control branches back to step **104** for menu selection and a new database server connection is made as appropriate. This process is repeated for each form. When no further selections are desired, the TAS firmware is exited at step **116**.

FIG. **8** is a flow diagram illustrating a technique for processing a form (step **110**) to create a data transaction in accordance with the invention. As illustrated, the process of FIG. **8** starts at step **118** and initializes a transaction buffer **97** at step **120** for storage of the data transaction as it is being created. In other words, if there is a form for the requested application, it is moved from form/menu memory **96** to the transaction buffer **97**. If the requested form is not present in form/menu memory **96**, an error message may be sent or a request may be sent to database server **28** for a download of a data stream containing the parameters for the requested form. Preferably, transaction buffer **97** is at least as large as the largest data transaction and serves as an assembly area for the data transaction. Preferably, read and write buffers are formed so that transmit and receive buffers to/from modem interface **78** are available. Of course, transaction buffer **97** may be made larger for this purpose.

Once the transaction buffer **97** is initialized at step **120**, the display screen **20** is cleared and the selected form is initialized to its first page at step **122**. The first page is then presented to the display screen **20** at step **124**. At step **126**, the user completes the form page on a field by field basis using any of the data entry techniques described above and the field controls of FIGS. **9** and **10**.

The transaction buffer **97** collects the data associated with the form presented to the user on display screen **20** and contains appropriate locations for each separate data element. Upon completion of the data transaction, the contents of the transaction buffer **97** are transferred to the appropriate database server(s) **28** via modem or via wireless, preceded by a set of codes (field **44**, FIG. **2**) which identify the type of data transaction and followed by a string of process identifiers for the database server(s) **28** to use in its programs in creating additional transactions and in storing the data and all ancillary data transactions in the regular file format of the database **30** associated with the database server(s) **28**. As a result, the data transaction created in the transaction buffer **97** has a one-to-many relationship to the data stored in the database **30**.

If the user decides to abort the processing of a form at any time (step **128**), the form processing routine is exited at step **129**. Otherwise, it is determined at step **130** whether the user wishes to go back a page (for a multi-page form) to correct a data entry. If so, control returns to step **124** for presentation of the earlier page. If the user does not wish to examine or edit a previous page, it is determined at step **132** whether the current form has another page which has not been displayed for completion by the user. If the form has more pages, the routine moves to the next page at step **134**, and it is determined at step **136** whether the move to the next page was successful. If so, control returns to step **124** for presentation of the next page. Of course, the process of calling a subsequent page in a form or another form upon completion of a form can be dependent upon an automatic call of that page or form sequence or the ability to jump sequence (i.e., skip pages) depending upon a value in any one field that has been entered. In any event, if there are no more pages in the form or if the move to the next page was not successful, the end of the form is marked with a code and the transaction is saved at step **138** by sending the data transaction to the appropriate database server(s) **28** for storage in its associated database **30** and "explosion" for storage of data in other databases **40**. If it is determined at step **140** that the save was not successful because of a modem error and the like, control returns to step **122** and the process is repeated. If the data transaction was successfully saved, the form processing routine is exited at step **129** and the last menu used is presented (step **112**).

Optionally, stored procedures within any data transaction form (field **50**, FIG. **2**) are executed at the appropriate time within the flow of the form processing routine before it is exited. However, these processes may be deferred and performed by the database server **28** if needed.

FIGS. **9**(*a*) and **9**(*b*) together illustrate a flow diagram of a technique for completing and editing the fields of a form (step **126** of FIG. **8**). The field completion routine starts at step **142** and first determines at step **144** whether an abort or a valid page move request is pending. If so, the field completion routine is exited at step **146**. However, if no abort or page move request is pending, the field data for the first field of the transaction buffer **97** is entered at step **148**. As noted above, this field data may be entered via keyboard **68** or touch screen **64**, swiped in via magnetic card interface **70**, read in from a memory card via memory card interface **89**, read in via modem interface **78** from database server **28**, or designated by voice entry. Preedit processing of the field data is then performed at step **150**. Such pre-edit processing may include, for example, setting default values, performing calculations, establishing links to data in other files, looking up and writing data to files already linked to the present form, spawning another form, performing special updates of the display screen **20**, hiding fields from view by the user, and the like. Such pre-edit processing may also be used to

determine whether modifications or actions in the present field may invalidate an entry in another interrelated field. If so, appropriate measures are taken to update all affected fields or to prevent such problems by setting appropriate default values.

The field completion routine then checks for field errors at step **152** on the basis of the default values and the like set at step **150**. If there is no field error at step **152**, it is determined at step **154** whether the operator will be permitted to edit the field in the absence of a field error. If so, or if a field error was found at step **152**, the operator edits the field at step **156**. If the operator editing is bypassed, control proceeds directly to post-edit processing at step **158**, which performs essentially the same functions as pre-edit processing step **150** except that the data may be specially validated. The field is then checked yet again at step **160** for a field error. If a field error is found at step **160**, control returns to step **144** for processing the next field or exiting, as appropriate.

If no field error is found at step **160**, it is determined at step **162** whether the generic field validation routine of step **164** (FIG. **10**) is to be skipped. If so, control proceeds to step **166**, where the field is once again checked for a field error. However, if generic field validations are desired, control passes to the routine of step **164** (FIG. **10**). If no field error is found at step **166**, the field is saved to the transaction buffer **97** at step **168** and the updated field value is painted on the display screen **20** at step **170**. If the user then desires to check a previous field at step **172**, control passes to a previous field at step **174** and the field completion routine is repeated for the previous field. However, if no previous field is to be checked and if it is determined at step **176** that a further field is present, control passes to the next field at step **178** and the field completion routine is repeated for the next field. This process repeats until the last field is completed and the routine exits at step **180**. Control then returns to FIG. **8** for processing a different page of the form.

Each form may be processed in one or more modes. In the input mode, described above, the data transaction is created and transmitted to the database server **28**. However, in edit mode, upon entering the ID of a particular record, that record is read from an external database **30** or **40** into transaction buffer **97** for editing. Preferably, a record of the edits is maintained to provide an audit trail. In view mode, upon entering the ID of a particular record, that record is similarly read from an external database **30** or **40** into transaction buffer **97** but for display only. Finally, in delete mode, an entire record can be deleted from the database **30** or **40** if the user has proper security clearance.

FIG. **10** illustrates how the TAS firmware validates the fields of each data transaction. As shown, the field validation routine starts at step **182** and first determines at step **184** what field type is present. If the present field is an alphanumeric field, control passes to step **186** where the field defaults are processed. It is then determined at step **188** whether the user knows the values allowed for this field. If not, and data is to be implanted in that field, an implant table is searched at step **190**. A "?" may be used by the operator to indicated that he or she does not know the values allowed for this field and wishes to search the implant table. A list of possible values are then called up that match the data entered thus far. From this list, the operator can scroll the list and select the value that will complete the data entry. However, if the value is not found in the list, a field error is generated at step **192** and the field validation routine is exited at step **194**. If the value is found in the list, control passes to step **200**.

On the other hand, if at step **188** it is determined that data need not be added (implanted) into the present field, control skips to step **196**, where it is determined whether the present field type is a field which sets up an event in which the present field (along with its form) can be linked to any record of any file or files (one to many) of any database for the purpose of data verification and/or data extraction. If so, control passes to step **198**, where the data from the present field along with any other data previously gathered is used to make the desired link. As in the data implant step **188** noted above, the user may enter a "?" to get the information needed to make this link. If the data for the link is not found, a field error is issued at step **192** and the field validation routine is exited at step **194**. However, if the data for the link is found, the field is checked for blanks at step **200** and a field error is issued at step **192** if blanks are present in the field but are not allowed. If no blanks are found in the present field, or if blanks are found but are allowed, the field validation routine is exited at step **202**.

If it is determined at step **184** that the present field is a numeric field, the field is checked at step **204** to determine if the character set is valid. If so, the precision of the numbers is adjusted at step **206**, as necessary, and the range and scope of the numbers are checked at step **208** to make sure the field entries satisfy the boundary conditions (e.g., no dividing by zero). If the character set is not valid at step **204** or the range and scope of the numerals is not valid at step **208**, a field error is issued at step **210** and the data validation routine is exited at step **212**. Otherwise, the field validation routine is exited at step **214**.

If it is determined at step **184** that the present field is a date/time field, the field is checked at step **216** to determine if the character set is valid. If not, a field error is issued at step **210** and the field validation routine is exited at step **212**. Otherwise, a routine of the TAS firmware checks the date/time entry at step **218** to determine if it has the correct format by performing range checking and the like. If the date/time entry does not have the correct format, a field error is issued at step **210** and the field validation routine is exited at step **212**. Otherwise, it is determined at step **220** whether the present field contains a date. If not, the data validation routine is exited at step **221**. If so, the date is checked at step **222** so see if it contains a weekend, and, if so, checks at step **224** whether a weekend date is an acceptable reply for this field. It is then determined at step **226** whether the calendar file is to be checked, and if so, the calendar file is checked at step **228** to see if the date is valid (e.g., not a February 30 and the like). Finally, it is determined at step **230** whether a warning date has been exceeded, and if so, a field error is issued at step **210** before the field validation routine is exited at step **212**. Otherwise, the field validation routine is exited at step **221**.

Those skilled in the art will appreciate that, in order to maintain security, the TAS firmware may also present a security form for password entry to the user. The security form and ID of the transaction entry device **12** is then encrypted and transmitted to the database server **28** associated with the particular data transaction assembler **18**. Transaction controller **36** of that database server **28** will then act as the transaction controller for that data transaction assembler **18** and will check passwords and the like during operation to make certain that data security is not breached. Database servers **28** may disable a data transaction assembler **18** if unauthorized use is attempted. In this manner, only the appropriate person may view each menu. Of course, a different number of security levels and different executive menus may be presented as desired, all under control of the transaction controller **36**.

C. Database Server **28**

As noted above, the database server **28** acts as a vehicle for separating data transactions created by the data transaction assembler **18** into the component parts thereof which may be stored directly in one or more databases **30** and **40**. In other words, the database server **28** explodes the initial data transaction into data transactions for many different files for updating records in the files, and the like. Also, the database server **28** may be virtual as well as real, exist in a single machine or in multiple machines, in whole or in part.

Generally, the database server **28** handles any and all data transactions received, manipulates data in the data transactions, spawns or starts processes or reports requested by a data transaction, and explodes the received data transactions into all sorts of data transactions that were spawned by the initial data transaction. Database server **28** can also update values in existing records and can switch to a process for processing values in the records as necessary. In this manner, a single data transaction can define actions causing multiple files to be updated. Database server **30** also handles requests from the data transaction assembler **18** and processes them as needed. Such requests may include data I/O requests, data locking and unlocking, report processes, and requests for new forms or menus. Those skilled in the art will appreciate that database server **28** maintains the one-to-many relationships that exist between the user and the system of the invention, the one-to-many presentations to the user and files in the databases **30** and **40**, and the one-to-many data transactions and the ancillary records, updates, and postings as may be required to diverse computer files of numerous databases **40**, the transaction entry device **12** and the database servers **28**.

As noted above, transaction buffer **97** collects the transaction data associated with the form presented to the user via display screen **20**. The transaction buffer **97** is the image of the data transaction with appropriate locations for each separate data element. The contents of the transaction buffer **97** are transferred to the database server **28** via modem interface **78** or via RF transceiver **90**, preceded by a set of codes **44** (FIG. **2**) which identify the type of transaction followed by a string of process identifiers for the database server **28** to use in its programs, in creating additional data transactions, and in storing the data and all ancillary transactions within the database **30** in the regular file format of the database **30**. In other words, the database server **28** determines what type of action to take based on the type of data transaction received, "explodes" a data transaction into a plurality of other data transactions for transmission to other databases, as appropriate, and converts the data for its associated database **30** into the proper file format. Of course, each database server **28** is different from each other database server **28** since it will handle different types of data transactions, have different operating system characteristics, and different file conversions to make in accordance with the file formats of its associated database **30**. For example, the database server **28** may operate under an operating system such as Unix, Windows, or DOS, where the operating system provides the database server **28** with links to the hardware functions normally handled by an operating system. Preferably, the database server **28** also operates with menus, forms, and the like in the same fashion as the data transaction assembler **18** except that it stores the data transactions in its associated database **30** as transaction files.

As just noted, the purpose of the database server **28** is to process the data transaction from the data transaction assembler **18** and to either explode the data transaction into all of its related components for storage, to handle the storage of items from the explosion process, to store the data transaction itself for reference purposes, and to act as a supplier of information to the data transaction assembler **18** in response to requests during the creation of the data transaction and during the downloading of parameters for menus and forms to the data transaction assembler **18**. If desired, the database server **28** can also supply information back to the data transaction assembler **18** after a data transaction is received or can initiate a process leading to the delivery of a report, data, or menu to the data transaction assembler **18**. In addition, the database server **28** and data transaction assembler **18** can reside on the same machine so long as the database server's operating system recognizes the TAS firmware or the TAS firmware is modified for use with the operating system of the database server **28**.

D. Applications of the Invention

As outlined above, the present invention includes a point of transaction device which presents a menu to a user from which an option is selected. A form tailored to the selected option appears for guiding the user through data entry. The full details of the data transaction are captured as data is entered by the user. Modem interaction with a central database(s) or a user database(s) allows for interaction for help and verification of certain entered data. The completed transaction is then transmitted to the central or user database for further processing and storage. Data input can also be provided via a swipe card or smart card, from data received from any database accessible via the modem interface, or other known methods.

A data transaction system of this type may be used for many applications. For example, in a first, presently preferred, application, the transaction entry device **12** is located in a medical office for entry of patient data. In this application, a swipe card identifies the patient, a smart card identifies the doctor, and the modem connection allows the entire claim transaction to be entered and transmitted to the insurance companies for processing. The patient records may also be automatically updated and prescriptions created, given to the patient, transmitted to the pharmacist, and transmitted to the payor and patient record. Patient instructions such as special diets, exercises, treatments, appointments, and the like may be printed from the data transaction form at the doctor's central computer. In addition, a video image or picture provided via video input **74** and compressed by data compression circuitry **75** permits an image of a medical condition such as a rash to be appended to the data transaction (in miscellaneous processing field **50** of FIG. **2**) for transmission with the patient's name, the date, a description of patient symptoms, and the like. Similarly, a recorded heartbeat may be appended to the end of the data transaction for transmission with the patient data.

The data transaction entry system of the invention also has numerous home uses. In a preferred home use, the transaction entry device is used for performing bank transactions from the home. In this case, forms would be made available by the bank for different types of bank transactions. These forms would then be downloaded to the transaction entry device in the customer's home and used in creating and transmitting data transactions to the bank computer for off-line processing.

As another example, the user may dial-up to a 900 number to get an interface to a central database which will download codes into TAS PROM **95** or form/menu memory **96** which enable the generation of infrared signals at certain frequencies. The user needs only to specify the type, make and

model of any electronic device to be controlled in order to get the desired code. Then, to operate any electronic device in the home, the user would be directed by menu prompts. The transaction entry device **12** would then emit an infrared signal via infrared transceivers **80** to operate the electronic device, initiate a call via modem for a broadcast program, or initiate timed requests for video recording, turning the video recorder on and off, and the like.

For other home uses, the transaction entry device **12** may also initiate, via menu prompts, sequences for turning on and off various household devices including alarm systems, coffeemakers, and the like. In this mode, the transaction entry device **12** may receive an RF or infrared signal indicating that a burglar or fire alarm has been activated and call up a form for calling the police or fire department, as appropriate. A call to the transaction entry device **12** may then be used to turn off the burglar or fire alarm by changing a field in a form which instructs the infrared transceiver **80** or RF transceiver **90** to send an appropriate control signal to the alarm device. This feature may also be prompted from a car phone via remote initiation of the form performing this function.

The transaction entry device **12** may also control all household telephone use as well as controlling the answering machine and keeping a telephone transaction log. The user may also pay household bills by completing an appropriate form and transmitting the form to a payee such as a credit card company, a bank, and the like. In short, the transaction entry device will permit the owner to connect to a remote database without owning a conventional computer system with an operating system and the like.

For personal applications, the transaction entry device **12** may be used to initiate a facsimile transmission, to provide telephone lists with automatic dialing upon selection, to provide expense accounts, personal scheduling, tax record keeping, and the like, and to provide direct access to travel information. For example, the database server **28** may be an airline reservations system. In this application, the data transaction assembler **18** dials the modem of the airline reservations system when the user requests data entry into an airline reservations form available at the user's transaction entry device **12**. The data transaction device **18** modems the database server **28**, and the operating system of the database server **28** selects interface programs for the airline reservations system. The interface programs call the database servers **38** of the airlines, retrieve the appropriate menu from database **40**, and modem the menu to the data transaction assembler **18**. The data transaction assembler **18** then displays the airline reservations menu on its display screen **20** for completion and transmission back to the airline reservations database server for processing. The swipe card may be used to provide credit card payment information and may be updated by permitting the data transaction assembler **18** to write to the swipe card. The user may also access frequent flyer club and mileage data, special offers on hotels, cruises and other travel, and the like.

In another home (or business) use, the transaction entry device **12** may be used to eliminate conventional phone mail greetings by enabling the caller's transaction entry device **12** to read in a set of visible menus from the called party's voice mail menu so that the calling party may select the desired options using a visible menu rather than a voiced menu. In other words, the caller would not have to wait through the litany of voiced phone mail options before making a selection and could make the desired selection right off of his or her own display. This would be accomplished by selecting a process from the menu of the transaction entry device **12** which will create a "visible" menu. When such a process is selected, the telephone electronics **14** or modem interface **78** makes a telephone connection to a remote phone mail system. Once the connection is made, the data transaction assembler **18** sends a data request for a visual representation of the phone mail menu of the remote phone mail system via the telephone connection to the remote phone mail system. A data stream containing the visual representation of the phone mail menu from the remote phone mail system is then returned via the telephone connection and stored in form/menu memory **96** and presented to display screen **20** of the transaction entry device **12** for selection using the techniques described herein. When menu items are selected from the "visible" voice mail menu, the data transaction assembler **18** creates a data transaction indicating which menu item was selected and sends the data transaction to the remote phone mail system via the telephone connection. Based on the menu selection, the remote phone mail system then returns a data stream containing a visual representation of the next phone mail menu via the telephone connection for storage in form/menu memory **96** and display on display screen **20**. This process is repeated until the calling party is required to leave a message or the called party is reached. Such a system would be particularly helpful for interacting with voice mail systems, such as those at government offices, where numerous options are presented for selection.

Those skilled in the art will appreciate that the invention is unique by virtue of its ability to generalize applications to forms so that no code need to be written to implement a particular function. However, if code is needed or if multimedia data is to be part of a data transaction, it can be attached to a form which is stored as a parameter stream in a stream of data. Also, though the transaction entry device **12** has been described as a computer workstation, it can also be used in conjunction with an optional off-line storage device as a self-contained workstation and database unit independent of traditional operating systems. The transaction entry device **12** can also be used with an additional optional plug in as a network server or as a user interface in a network docking station.

Those skilled in the art will also appreciate that the foregoing has set forth the presently preferred embodiments of the invention but that numerous alternative embodiments are possible without departing from the novel teachings and advantages of the invention. Accordingly, all such modifications are intended to be included within the scope of the appended claims.

I claim:

1. A system for entering transaction data into a remote database, comprising:

a data input device;

a display;

a data transaction terminal including a microprocessor, a form memory which stores a plurality of menus and forms for presentation to a user, and a form driven operating system which controls a process implemented by said microprocessor to present to said display for each process at least one form stored in said form memory as data streams, said at least one form being selected by said user from one of said menus using said data input device, said one menu providing said user with an option of selecting at least one of said at least one form, another menu, and an updating process, each form eliciting data input of a desired transaction type into said data input device by said user and including at least one prompt customized to said

desired transaction type, wherein said process implemented by said microprocessor is changed by changing said at least one form, and wherein when said user selects said updating process from said menu, data streams are downloaded to said form memory to update said menus and forms in accordance with said desired transaction type, said data transaction terminal further including means for formatting at least said data input by said user in response to said at least one prompt into a data transaction for transmission to said remote database; and

a database server associated with said remote database which receives said data transaction, creates from said data transaction, depending on said desired transaction type, at least one additional data transaction containing data for a particular record in said remote database, and stores said at least one additional data transaction in said particular record.

2. A system as in claim **1**, further comprising a plurality of remote databases, wherein said database server further creates from said data transaction, depending on said desired transaction type, at least one ancillary data transaction containing data for a particular record in one of said plurality of remote databases besides said remote database and stores said at least one ancillary data transaction in said particular record.

3. A system as in claim **2**, wherein said form driven operating system includes means for sending a data request to said database server, said database server accessing data corresponding to said data request in at least one of said remote databases and returning one of data responsive to said data request, a list of options for selection by said user, a value calculated from data contained in said data request, and a data report.

4. A system as in claim **1**, wherein said form driven operating system comprises a transaction assembly server (TAS) which presents said data streams to said microprocessor for display on said display, and said formatting means comprises a transaction buffer which stores said data input into said data input device by said user in response to said at least one prompt until said data transaction is completed for transmission to said remote database.

5. A system as in claim **4**, wherein said data transaction terminal further comprises a modem, a telephone and two telephone line connections, one for connecting said telephone to a telephone network, and one for providing a modem connection among said modem, said TAS, and said database server.

6. A system as in claim **5**, wherein said data transaction terminal further comprises a mode switch for selecting a telephone mode in which said data transaction terminal operates exclusive of said TAS or a transaction entry mode in which said TAS operates exclusive of said telephone.

7. A system as in claim **4**, wherein said data transaction terminal further comprises a modem, a telephone, a telephone line connection, and means for selectively connecting said telephone to a telephone network and said TAS to said database server via said telephone line connection.

8. A system as in claim **7**, wherein said selectively connecting means comprises a mode switch for selecting a telephone mode in which said data transaction terminal operates exclusive of said TAS or a transaction entry mode in which said TAS operates exclusive of said telephone.

9. A system as in claim **1**, wherein said database server comprises a modem, a data transaction queue for storing data transactions received from said data transaction terminal, and a transaction controller which processes the received data transactions to extract physical relationships of data of said data transactions with records in said remote database.

10. A system as in claim **1**, wherein said one menu further contains a remote process option, and when said user selects said remote process option from said one menu, data streams are downloaded via a modem to said form memory, said data streams containing control data for implementing functions designated by said selected remote process option.

11. A system as in claim **10**, wherein said data transaction terminal further comprises an infrared transceiver and said control data comprises data for controlling a wavelength of energy emitted by said infrared transceiver.

12. A system as in claim **10**, wherein said data transaction terminal further comprises a phone list memory for storing a phone list and said control data comprises data for updating said phone list.

13. A system for entering transaction data into a plurality of remote databases, comprising:

a data transaction terminal for capturing a data transaction having a one-to-many relationship to records of said plurality of remote databases, said data transaction terminal including a microprocessor, a form memory which stores a plurality of menus and forms for presentation to a user, and a form driven operating system which controls a process implemented by said microprocessor to present to said user for each process at least one form stored in said form memory as data streams, said at least one form eliciting data input of a desired transaction type into said data transaction terminal by said user, said data streams of said at least one form including at least one prompt customized to said desired transaction type, a format field which identifies said desired transaction type of said at least one form, a data entry field including said at least one prompt and spaces for said data input by said user in response to said at least one prompt, and a processing field for appending data particular to said desired transaction type, said data transaction terminal further including means for formatting at least said data input by said user in response to said at least one prompt into said data transaction for transmission to at least one of said plurality of remote databases;

a first set of database servers associated with a first set of remote databases of said plurality of remote databases, said first set of database servers receiving said data transaction, creating from said data transaction a plurality of ancillary data transactions having a one-to-one relationship to said records of said plurality of remote databases, and storing said ancillary data transactions in designated records of said first set of remote databases; and

a second set of database servers associated with a second set of remote databases of said plurality of remote databases, said second set of database servers receiving certain of said plurality of ancillary data transactions, and creating from said certain ancillary data transactions additional data transactions which are stored in application specific records of said second set of remote databases in accordance with said desired transaction type of said at least one form.

14. A system as in claim **13**, wherein each database server of said first and second set of database servers comprises a modem, a data transaction queue for storing data transactions, and a transaction controller which processes a received data transaction to extract physical relationships of data of said data transactions with records of a remote database associated with said each database server.

15. A system as in claim 13, wherein said data particular to said desired transaction type includes at least one of audio and video data.

16. A data transaction terminal for providing data transactions to a remote database server which stores records in an associated database, comprising:

a data input device;

a display;

a telephone circuit;

a data transaction assembler including a microprocessor, a form memory which stores a plurality of menus and forms for presentation to a user, and a transaction assembly server (TAS) which controls a process implemented by said microprocessor to present to said display for each process at least one form stored in said form memory as data streams, said at least one form being selected by said user from one of said menus using said data input device, said one menu providing said user with an option of selecting at least one of said at least one form, another menu, and an updating process, each form eliciting data input of a desired transaction type into said data transaction assembler by said user and including at least one prompt customized to said desired transaction type, wherein said process implemented by said microprocessor is changed by changing said at least one form, and wherein when said user selects said updating process from said menu, data streams are downloaded to said form memory to update said menus and forms in accordance with said desired transaction type, and means for formatting at least said data input by said user in response to said at least one prompt into a data transaction for transmission to said remote database server; and

a mode switch for selectively connecting said telephone circuit to a telephone network in a telephone mode and said data transaction assembler to said remote database server in a data transaction entry mode.

17. A terminal as in claim 16, wherein said formatting means comprises a transaction buffer which stores said data input into said data input device by said user in response to said at least one prompt until said data transaction is completed for transmission to said remote database.

18. A terminal as in claim 16, further comprising a modem and two telephone line connections, one for connecting said telephone circuit to said telephone network, and one for providing a modem connection among said modem, said TAS, and said remote database server.

19. A terminal as in claim 16, further comprising a modem and a telephone line connection, said mode switch selectively connecting said telephone and said TAS to said telephone line connection.

20. A terminal as in claim 16, wherein said one menu further contains a remote process option, and when said user selects said remote process option from said one menu, data streams are downloaded via a modem to said form memory, said data streams containing control data for implementing functions designated by said selected remote process option.

21. A terminal as in claim 20, further comprising an infrared transceiver, said control data comprising data for controlling a wavelength of energy emitted by said infrared transceiver.

22. A terminal as in claim 20, further comprising a phone list memory for storing a phone list, said control data comprising data for updating said phone list.

23. A terminal as in claim 17, wherein said TAS presents one of said menus to said user for selection, said one menu containing pointers to a plurality of forms, and upon selection of said at least one form from said menu by said user, said TAS initializes said transaction buffer and presents said at least one form to said display on a page by page basis for entry of said input data by said user.

24. A terminal as in claim 16, wherein said TAS processes said input data as it is entered in response to each prompt to determine if said input data satisfies predetermined conditions for input data entered in response to each said prompt.

25. A terminal as in claim 16, wherein said TAS sends a data request to said remote database server when said user requests assistance in replying to a prompt and inserts reply data from said remote database server into said data transaction in response to said prompt.

26. A terminal as in claim 16, wherein said TAS comprises means for creating from said data transaction, depending on said desired transaction type, at least one ancillary data transaction containing data for a particular record in said associated database and storing said at least one ancillary data transaction in said particular record in said associated database.

27. A terminal as in claim 26, wherein said ancillary data transaction creating means further creates from said data transaction, depending on said desired transaction type, an ancillary data transaction containing data for an application specific record in a secondary database and sends said ancillary data transaction to said secondary database for storage of said ancillary data transaction in said application specific record.

28. A terminal as in claim 16, wherein said data transaction assembler includes means for sending a data request to said remote database server, said remote database server accessing data corresponding to said request in said associated database and returning one of data responsive to said request, a list of options for selection by said user, a value calculated from data contained in said data request, and a data report.

29. A terminal as in claim 16, wherein said data input device comprises at least one of a touch screen associated with said display, a telephone numeric keypad, an alphanumeric keyboard, a memory card reader, and a magnetic card reader.

30. A terminal as in claim 29, wherein said alphanumeric keyboard comprises a retractable keyboard which retracts into a housing of said data transaction terminal.

31. A terminal as in claim 16, further comprising a video input terminal for receiving input video data and a video output terminal for providing output video data to a video monitor.

32. A terminal as in claim 31, further comprising a data compression circuit for compressing said input video data prior to including said input video data in a data transaction and a data decompression circuit for decompressing output video data prior to display on said video monitor.

33. A terminal as in claim 16, further comprising a computer I/O port for receiving input data from a computer device and providing output data to at least one of said computer device and a printer.

34. A terminal as in claim 16, further comprising a RF transceiver for providing a wireless connection between said data transaction terminal and a data processing device.

35. A terminal as in claim 16, further comprising a battery for providing power to said data transaction terminal for portable operation.

36. A terminal as in claim 16, wherein said data input device comprises a voice recognition circuit for accepting data input selections annunciated by said user.

37. A terminal as in claim 16, further comprising a voice synthesizer responsive to said data transaction for audibilizing a portion of said data transaction to said user.

38. A terminal as in claim 37, further comprising a voice recorder for recording at least one of said audibilized portion of said data transaction when in said data transaction entry mode and voice input from a called party when in said telephone mode.

39. A method of entering transaction data into a remote database using a data transaction terminal, comprising the steps of:

storing a plurality of menus and forms in a form memory of a form driven operating system of said data transaction terminal, each form including at least one prompt customized to a desired transaction type;

said form driven operating system controlling said data transaction terminal to accept input data of said desired transaction type using control data comprising at least one of said forms from said form memory;

a user selecting one of said menus using a data input device and said user selecting from said one menu at least one of said at least one form, another menu, and an updating process for further processing;

if said updating process is selected from said one menu by said user, downloading data streams to update said menus and forms in accordance with said desired transaction type; and

if said at least one form is selected from said one menu by said user, said form driven operating system presenting to a display a form for eliciting data input of said desired transaction type from said user, said user inputting data in response to said at least one prompt of said form using said data input device, and said form driven operating system formatting at least said input data from said user into a data transaction for transmission to said remote database and transmitting said data transaction to said remote database.

40. A method as in claim 39, comprising the additional steps of:

receiving said data transaction at said remote database;

creating from said data transaction, depending on said desired transaction type, at least one additional data transaction containing data for a particular record in said remote database; and

storing said at least one additional data transaction in said particular record.

41. A method as in claim 40, comprising the additional steps of:

creating from said data transaction, depending on said desired transaction type, at least one ancillary data transaction containing data for a particular record in an ancillary database different from said remote database; and

storing said at least one ancillary data transaction in said particular record in said ancillary database.

42. A method as in claim 39, comprising the additional steps of:

sending a data request to a database server of said remote database; and

said database server accessing data corresponding to said data request in said remote database and returning one of data responsive to said data request, a list of options for selection by said user, a value calculated from data contained in said data request, and a data report.

43. A terminal as in claim 28, wherein said remote database server is a remote phone mail system and said telephone circuit makes a telephone connection to said remote phone mail system, said data request being sent via said telephone connection and including a request for a visual representation of selection options of a phone mail menu of said remote phone mail system, and, in response to said data request, said remote phone mail system returning via said telephone connection a data stream containing said visual representation of said selection options of said phone mail menu, said visual representation of said selection options of said phone mail menu being presented to said display by said data transaction assembler for selection by said user using said data input device, and said data transaction assembler further sending data to said remote phone mail system via said telephone connection indicating which selection option was selected from said phone mail menu by said user.

44. A terminal as in claim 43, wherein said remote phone mail system returns a data stream containing a visual representation of selection options of a next phone mail menu via said telephone connection in response to said data indicating which selection option was selected from said phone mail menu by said user.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 5,805,676
DATED : September 8, 1998
INVENTOR(S) : Rocco L. Matino

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

Column 10,
Line 42, change "110-13" to -- 110-113 --.

Column 12,
Line 12, change "600" to -- 60° --

Column 16,
Line 53, "The data stream entered etc." does not start a new paragraph.

Signed and Sealed this

Twenty-seventh Day of May, 2003

JAMES E. ROGAN
Director of the United States Patent and Trademark Office