A PALM COMPUTER BASED MOBILE INFORMATION SYSTEM FOR CLINICAL HEADACHE RESEARCH

A Thesis Presented to

The Faculty of the

Fritz J. and Dolores H. Russ
College of Engineering and Technology

Ohio University

In Partial Fulfillment

of the Requirement for the Degree

Master of Science

by

Yi Chen

March, 2001
Acknowledgements

So many people have contributed to the completion of this thesis. Dr. Ken Holroyd, professor of psychology at Ohio University, initialized this project with a NIH headache research fund. We have been working together on this project ever since the beginning. I can’t express the importance of his contributions to this effort. Dr. Jeffery Dill, professor of electrical engineering and computer science and my academic advisor, has offered his continuous support during my studies at Ohio University. He taught me how to do research and overcome technical and mental obstacles. I have enjoyed all the time spent with him. Dr. Connie Cottrell in Ortho/Neuro Corporation (Columbus, Ohio), Mike Stensland and other people in Headache Research Center of Ohio University have provided constructive criticism and helped to test the system. I feel very grateful for their help.

I owe a lot to my family. My wife Sherry has always stood by me. My mother-in-law came such a long way from Beijing to help us. I also want to mention my parents and sisters. They have been the spiritual force all these years that has helped me move forward. Thanks to all of you and I want to share all my accomplishments with you.
# Table of Contents

List of Figures.........................................................................................vi

Part I Introduction...................................................................................1

1.1 Palm Operating System.................................................................2

1.2 Palm Hardware..............................................................................4

1.3 Palm Software...............................................................................6

1.4 Palm Software Development Tools.............................................6

1.5 Palm Economy..............................................................................8

1.6 NIH Headache Research Project...................................................9

Part II Software Development..............................................................12

2.1 Product Requirement......................................................................12

2.2 Use Case Analysis.........................................................................15

2.2.1 Palm Platform Application.....................................................15

2.2.2 Desktop Platform Application................................................17

2.3 Development Tools for Diary......................................................19

2.4 Diary Information System Data Flow.........................................20

2.5 Palm Diary Control Flow..............................................................20

2.6 Desktop Diary Database Schema and Data Flow.......................22

2.7 Database Operations in the Desktop Database.........................24

2.8 Data Export and Import...............................................................26

2.9 System Verification.......................................................................27
List of Figures

Figure 1.1 Palm OS architecture ................................................................. 3
Figure 2.1 Diary information system data flow ........................................... 20
Figure 2.2 Palm diary control flow .............................................................. 21
Figure 2.3 Diary database schema on the desktop side ............................... 23
Figure 2.4 Diary flow in the desktop database ............................................ 24
Figure 2.5 Form relationships if the desktop database application .............. 26
Figure 3.1.1: The letter “D” icon launches the Diary in a Palm .................... 31
Figure 3.1.2: Patient info configuration form .............................................. 31
Figure 3.1.3: The starting form for the Diary program ............................... 32
Figure 3.1.4: Form 2 for a male patient ..................................................... 32
Figure 3.1.5: Form 2 for a female patient .................................................. 32
Figure 3.1.6: Form 3 starts the headache recording process ....................... 32
Figure 3.1.7: Form 4 for a record w/o ending time info ............................. 33
Figure 3.1.8: Form 5 for recording headache detail ................................... 33
Figure 3.1.9: Form 6 for recording headache symptom info ....................... 33
Figure 3.1.10: Form 7 for recording headache medication usage .................. 33
Figure 3.1.11: Form 8 for part one of question A ...................................... 34
Figure 3.1.12: Form 9 for part two of question A ...................................... 34
Figure 3.1.13: Form 10 for part one of question B .................................... 34
Figure 3.1.14: Form 11 for part two of question B .................................... 34
Figure 3.1.15: Form 12 for part one of question C.................................35
Figure 3.1.16: Form 13 for part two of question C.................................35
Figure 3.1.17: Form 14 for part one of question D.................................35
Figure 3.1.18: Form 15 for part two of question D.................................35
Figure 3.1.19: Form 16 for headache record summary............................36
Figure 3.1.20: Form 17 for daily information recorded............................36
Figure 3.1.21: Form 18 to remind the user of missing days.....................36
Figure 3.1.22: Form 18 to exit the program........................................36
Figure 3.2.1: Diary database launcher...............................................37
Figure 3.2.2: Diary database main form............................................37
Figure 3.2.3: Form to upload data from the Palm into the database............38
Figure 3.2.4: Diary database form to create a result chart or table.............38
Figure 3.2.5: Diary database form to display analysis results in a table.......39
Figure 3.2.6: Diary database form to display analysis results in a chart.....39
Part I   Introduction

Although we live in a digital era and computer technology is widespread in today’s world, it has not reached the point of accommodating all needs. We observe a UPS driver scans the delivered merchandise to record shipping information. That same day, we might see a physician write down patient information on a piece of paper and wait for that to be typed into the computer. This is an awkward situation when we think about how important it is for physicians to record information correctly and access the existing information quickly. It would be extremely valuable if they had a mobile computer to do their job as the UPS driver does. Fortunately, this situation is changing now as the Palm computer is being recognized as an enterprise solution to the healthcare industry. A Palm is a palm-size computer running the Palm operating system. Currently, it holds more than 75% of the mobile computing market. The healthcare industry has started to adopt this technology. In some hospitals, physicians use applications running on Palm computers to prescribe medication for patients and record billing information. The work presented in this thesis takes these solutions one step further. I developed a Palm-based application used by patients, instead of medical doctors, to actively record their headache treatment information. In addition, I developed a back-end database application to provide data analysis and a conduit application to transfer data from Palm to database.
1.1 Palm Operating System

Palm OS has a flexible architecture for innovative solutions. The Palm OS platform consists of five primary components as listed below. Figure 1.1 provides an overview of the internal architecture, including both hardware and software layers.

1. Palm OS® software;
2. Reference hardware design;
3. HotSync® conduit data synchronization technology for one-button synchronization;
4. Platform component tools including an API that enables developers to write applications;
5. Software interface capabilities to support hardware add-ons.

The current Palm operating system (OS) version is 3.5. Version 4.0 may be shipped in the beginning of the year 2001. One of the important features of the Palm OS is the event manager. The event manager stores events and handles them in the order they are received. The core part of a Palm-based application is all about event handling.

The hardware layer of a Palm computer has the same components as a regular computer except that it does not have a hard disk. Instead, all the information is stored in a memory heap. The memory is divided into two parts: storage heap and dynamic heap. All the permanent data is stored in the storage heap and all of the
runtime data is stored in the dynamic heap. A Palm application usually has a very short response time to user interactions since there is no data access process from the hard disk as in a regular computer. Another reason is that the Palm OS was designed to be simple and efficient. This is a different approach from the Windows CE device and is the reason Palm is so successful in the market place. The limitations of a Palm computer are the small amount of memory (usually 2-8MB) and the small screen (around 3” by 3”). However, we would not expect a mobile computer to handle tasks as a regular desktop computer does. A Palm computer also has limited data processing capability. The most popular CPU is the Motorola Dragonball of 20MHz. There is no doubt that the newer models of Palm computers can use faster CPUs, however, a faster CPU usually means more battery power consumption. Battery life is a very big concern for handheld devices since data can be lost when a battery goes out for a period of more than one minute. This is the main reason why
Windows CE disappeared from the market in less than two years. The marriage of a Windows OS with a handheld computer was not successful because a handheld computer is not designed to handle as heavy a load as a regular PC. Windows CE takes much longer to boot up and drains the battery so fast that a regular user has to replace the battery every week. Palm OS is different from the Windows CE operating system because its goal is to keep the environment simple without losing the major functionalities of handheld devices. It can provide an excellent user experience without requiring a fast CPU and large amounts of memory.

Since a Palm computer has a relatively small amount of storage space, data synchronization between a Palm and a desktop is required to provide data backup. This is the key feature that will enable Palm computers to be successful in remote and instant data exchange in a mobile computing environment. The blooming market for Palm computers has proved that they are the solution for mobile computing.

1.2 Palm Hardware

There are three major Palm computer manufactures, including Palm Computing Inc., HandSpring and Symbol Technology. All Palm computers, although of different brands, are Palm OS compliant. Palm Computing has released four series of handhelds: Palm M100 series, Palm III series (IIIe, IIIx, IIIxe and IIIc), Palm V series (V and Vx) and Palm VII series (VII and VIIx). Palm M100 series are the economic models and designed for personal use. Palm III series are for both business
and personal use and represent Palm’s initial product. Palm V series are smaller and thinner than other Palms. They use rechargeable batteries instead of the AAA batteries used by the others. The Palm VII series is unique because they provide wide area networking capability. A user can check email and surf the web by subscribing to the service provided by www.palm.net. A Palm VII device communicates directly with the base stations built by BellSouth. These stations are then connected to the server at www.palm.net. It is the server that connects the wireless network and the wired internet. A Symbol handheld computer is a larger size because of an embedded laser gun and barcode scanner. It is equipped with local area networking capability. Such a device can communicate with a base station within a distance of about two kilometers. HandSpring is a spin-off of 3Com before Palm Computing Inc. left 3Com. HandSpring’s product line is called “Visor”. There isn’t too much difference between a Visor computer and those manufactured by Palm Computing. The appendix at the end of this thesis presents some pictures of typical Palm computers.

All Palm computers include the following features:

1. Applications: Date Book, Address Book, To Do List, Memo Pad, Expense, Calculator, desktop e-mail connectivity, Security, Games, and HotSync® software for local and remote synchronization;

2. Modem-enabled and Internet-ready with TCP/IP software;
3. An infrared port (IR), allowing the user to beam data and applications to any other IR-enabled Palm OS® handheld, or connect to other IR-enabled devices such as cellular phones, pagers, and laptops.

1.3 Palm Software

As discussed before, an orphan application on a Palm computer is not a complete solution because data can be lost when the battery goes out. Each Palm application has to communicate with an application running on a local desktop computer or a remote server so that data can be backed up and manipulated on both platforms. A complete Palm application should contain three components: an application running on the Palm, an application running on a desktop or remote server and a conduit application to handle data exchange across platforms. The conduit can be located on a local desktop or a remote server. It is the bridge by which data can be transferred across platforms to achieve data synchronization. The Palm operating system supports three types of Palm files, including prc, pdb and pqa. The prc files are the executable files. The pdb files are database files managed by prc files. The pqa files are for web clipping application to display web pages on the Palm platform.

1.4 Palm Software Development Tools

Software development on the Palm platform presents some unique issues. The developer has a smaller screen to display user interface components, a slower CPU, limited memory space and no hard disk. In addition, a developer has to pay special attention to memory management in order to avoid memory leaks. To achieve a
reasonable user experience, it is advised to avoid intensive data processing on the Palm computer, since doing that requires a fast CPU and a fast CPU means more battery power consumption.

Currently there are two types of development tools available for Palm platform development. The official development tool, "CodeWarrior", previously owned by Metroworks Corporation, has been taken over by Palm Computing Inc. This is a C/C++ based development tool. The current version is 6.0 for both Windows and Macintosh platforms. However, CodeWarrior on a Macintosh is far more mature than the Windows version since it was originally created by Macintosh developers. Another C/C++ based package is the GNU C/C++. It is freeware and provides no graphical user interface (GUI) development environment. The second category of development tool is akin to Visual Basic and suitable for fast development of less complex application. These tools can only be used on a Windows platform. Satellite Forms is the tool provided by Puma Technology. It has rich functionality that enables fast development. Currently Satellite Forms has two editions, the standard and the enterprise edition. The latter supports network communication while the former does not. Another development package is Pendragon Forms. Pendragon provides fewer development resources, especially in user interface components. When comparing these two, Satellite Forms is a much more powerful tool than Pendragon Forms.

There are different development tools for conduit development in Windows and Macintosh environments. On the Macintosh platform, CodeWarrior can be used with...
the conduit development kit (CDK). On the Windows platform, Visual C++ can be used. Conduit development is a big topic and will not be discussed here in detail. Further information about conduit development can be found at www.palmos.com².

1.5 Palm Economy

Palm computers have infused the mobile computing market with fresh blood. They are being adopted as enterprise solutions in today’s market, especially in the healthcare and wireless fields. Currently Palm devices represent more than 75% of the mobile computing market. Microsoft is actively seeking handheld solutions to stay competitive in this field. Handheld devices are impacting the market in several areas by enabling PC-based applications to be mobile or by supporting new applications. The first area is convenience. The handheld device will release working professionals from cumbersome schedule, notes and address books in their daily life. Instead, this information is effectively organized into a single portable handheld computer and can be accessed and updated quickly. The second area is efficiency. The handheld computer provides a very convenient way to record, manipulate and access information routinely collected by sales people, home-visiting nurses, hospital doctors, lab operators and other technical and professional personnel. Companies such as VirtMed, MD Everywhere, and Ephysician are developing applications for the healthcare industry. An application from VirtMed, “Charge Capture”, has been used by several Boston based hospitals. One hospital claimed that Charge Capture saved 30% of the total revenue lost to human error. The most exciting part of a handheld device is its wireless capability. Compared with traditional wireless
services, the Palm computer based wireless service is more versatile. That is why the traditional wireless industry is quickly adapting itself to the Palm computing platform. One example is the SmartPhone by Qualcuum. It combines a Palm and a cell phone into a single device. A user can do much more with the SmartPhone than a regular cell phone. Nokia, Bell South and other wireless companies are joining this competition\textsuperscript{10}.

1.6 NIH Headache Research Project

In a typical clinical trial of headache therapy, patients record symptoms, medication use and impact of daily life on paper forms\textsuperscript{11}. This requires them to transfer data from paper forms to computers. Errors can occur during the labor-intensive transferring process as well as at the time the patients record information. A handheld recording process can catch some input errors, such as inconsistent information and data out of valid range, and eliminate the manual data transferring process. In addition, the handheld computer can identify the time the data was entered and thus determine if the data was entered retrospectively or concurrently.

The purpose of my work is to develop a user-friendly Palm computer-based information system that will allow patients to record relevant headache information. The system should be usable by individuals with little or no computer experience. A Palm application has the potential not only to improve the quality of data collected in clinical trials of new therapies for migraine and tension-type headache, but also to provide the user information that will enable them to more effectively manage their
headaches. This project is a subproject of a NIH funded clinical trial (NINDS NS32374-06 “Drug and non-drug treatment of severe migraine”).

This NIH clinical trial will be evaluating preventive drug (Propranolol-LA/Nadolol) and non-drug (Behavioral Migraine Management) therapies for frequent severe migraines. Patients with frequent and severe migraines at two research sites will receive state of the art acute therapies for migraine and be randomized to one of four treatments groups: Preventive medication, preventive medication placebo, behavioral migraine management plus preventive medication placebo and behavioral migraine management plus preventive medication. After a three-month dose adjustment treatment period, patients will be followed for 12 months. Although a number of different types of data will be collected, including neurologist evaluations, psychosocial tests evaluating quality of life and other variables, the primary outcome data comes from patients daily recordings of headaches, disability and medication use with the “Diary” program developed in this project. Thus, the quality of the trial results is dependent on the quality of data recorded in the “Diary” program.

The use of a Palm computer as a data collection tool by patients is a new approach in clinical research. It has the potential not only to improve the quality of data collection, but also to improve the quality of headache care. Patients are provided with important summary information about their headaches - thereby enabling them to use medications and/or behavioral headache management skills more effectively. Physicians are provided with quantitative information about the patients’ headache
problems, disabilities and medication usage. Future versions of the Diary might also incorporate algorithms that alert patients to problem behaviors (e.g., excessive medication use or failure to use the proper medication) or allow patients access to tailored headache management information or consultation with a nurse or physician via wireless connection to a remote server.

We were originally interested in the Palm VII because it allows our patients to submit their data remotely. However, some of our participants live in rural areas where wireless service for Palm VII is unavailable. We decided to have patients bring in the Palm computers and upload their headache information during their office visit. A pilot study has proved that a Palm IIIx with 4MB memory can hold data for more than 6 months. The application running on the Palm computer will be called "Diary" since we hope the patients will use this program on a daily basis. Accordingly, the database application on the desktop computer is named the "Diary Database".
Part II Software Development

2.1 Product requirements

Our product is an information system that includes a data collection application on the Palm device and a data processing application on a personal computer. Data from subsets of patients located on different PCs can be merged into a single master database. A patient will use the Palm application to record her or his headache information for one to six months. Data entry must be user-friendly so the application can be used by individuals who do not have computer skills. All the data collected by the patient should be safely stored and transferred into a PC-based database. Fast data analysis is required to provide instant feedback to the patient as soon as the data is uploaded into the database. The Palm application should collect the following information for each treatment period:

1. Daily use of preventive medication;

2. Menstrual starting date for female patients;

3. Number of times the patient contacts the Diary program each day;

4. If there is a headache, the following information is required to be recorded:
   a. Start time/date;
   b. End time / date;
   c. Headache type (Migraine, Tension, Other and Unknown headaches);
   d. Headache intensity (None, Mild, Moderate and Severe)
   e. Time and date when this record is entered into the Palm;
f. Associated symptoms (None, Mild, Moderate and Severe)
   i. Phonophobia (Sensitivity to sound);
   ii. Photophobia (Sensitivity to light);
   iii. Nausea;
   iv. Vomiting (Yes / No);

g. Headache medication (Number of doses):
   i. Study Analgesic,
   ii. Immitrex (Pill, spray or injection),
   iii. Maxalt (Melt or pill),
   iv. Anti-nausea (Reglan)
   v. Rescue medication;

h. Disability
   i. Non-paid work (missed hours and work hours with more than 50% impairment);
   ii. Paid work (missed hours and hours worked with more than 50% impairment);
   iii. Social/recreational activities (missed hours and hours of activity with more than 50% impairment);

5. Missed sleep;

6. If the patient forgets to record information on any given day, the Diary application will remind the patient to fill in the missing information. The
application will continue to remind the user until two weeks from the day of
the missing information;

7. The program should check for missing or inconsistent data and alert the user
to provide the missing data or to correct the inconsistency in the data entered.

On the desktop database side, the application should accept data from Palm
computers and store the data in the database. It also provides a way to manage
patients’ participation in the headache research project. When data is uploaded into
the database, summary reports are generated that allow for the comparison of
treatment periods for a patient or group. For example, information such headache
severity, date/time, number of days when a headache occurred, medication usage,
etc. should be readily available. The report should be in the form of a bar graph or
tabular summary. A unique report should be generated based on the user’s input.
Reports can represent individual patient numbers or group numbers, headache type,
treatment period and/or other relevant variables.
2.2 Use Cases Analysis

2.2.1 Palm platform application

1. Definitions

   a. Missing Days

      A day when the patient does not contact the Diary application as indicated by the variable “contact times”. Contact occurs when a user answers or reviews daily questions on medication usage and menstrual flow starting date for female patients.

   b. Headache Day

      A day inside the headache duration window. The headache duration window is the time elapsed between headache start date and headache end date.

   c. Number of Headache Days

      The number of days of a headache duration window. For example, a headache starting on 08/07/2000 and ending on 08/08/2000 is considered to be 2 headache days with no regard to the exact time during the start and end day.

   d. Severe Headache Day

      A headache day when any Immitrex or Maxalt is taken by the patient or the patient records a disability item.

   e. Incomplete Headache Record
A headache record without an ending date and/or time.

f. >50% Headache Impact

When a headache significantly affects a patient’s performance in daily activity.

2. Record information

The patient can record information about daily use of preventive medication, menstrual flow starting date, and headache specific information as required. If the user answers “no” to the question “Do you have a headache to record?”, he or she should be presented with the table summary of their existing records. After reviewing the summary, the user can exit the Diary program. However, before Diary quits, it checks the patients records for the past two weeks for missing days. If a missing day is found (as indicated by a zero contact times), the patient should be prompted to answer the questions for the missing day. The patient can dismiss the question by choosing not to answer it, thereby causing the application to quit, or can continue by filling in the missing information. The logic of the Diary application consists of a set of questions. Some of the questions contain two parts. The first part is answered with yes or no. If yes, the second part of the question will be presented to the user. If no, the second part is omitted. A user should not be allowed to change an existing record for the purpose of data integrity. However, if the user has not finished recording a complete record, he or she can switch between forms to change the entries.
3. View summary

A patient can view their headache information from the first day information was entered to the current day to monitor their headache treatment progress on the Palm computer. The patient should be able to see if each day during the treatment period is a headache day or not and if he or she took the preventive medication. The summary should also include the start date, number of headache days, headache type, headache severity, and medication (Maxalt & Immitrex) doses for each headache.

4. Continue an incomplete headache record

If a patient did not finish recording a headache with an ending date and time, the Diary application will not allow the patient to proceed with a new headache record until the patient chooses to record the headache ending information. Diary will not check the previous records for missing days if an incomplete headache exists.

2.2.2 Desktop platform application

1. Patient management

A database user should be able to add and remove patients from the database. If a patient is removed, all data associated with this patient should also be removed. The database user does not have the authority to change any patients treatment group number. The group number should only be accessible by an authorized person, who can then assign the group number to patients for research purposes.
2. Data uploading

During data uploading from the Palm device into the database, the Diary database application should check that the patient using the Palm currently exists in the database. Otherwise, the program should exit without doing anything to the database. During upload, personal information about the patient should be displayed for verification. If there are any incomplete records that exist in the uploaded data, the user will be alerted to either manually process the incomplete data or discard the updated data. Data processing can not proceed unless all incomplete records are fixed.

3. Data analysis

Summaries of individual patient data or averaged data for all patients in a treatment group should be displayed using either a tabular or graphical form. The data for each treatment period, no matter how many days it lasts, should be adjusted to fit into a 30-day time window for easy comparison of different treatment periods. For example, if we have a 40-day treatment period and the patient has taken 24 Immitrex pills, the number of pills should be adjusted to 18 in the summary table. Each table or graph should be based on the user’s choice of a summary of patient or treatment group, headache type and treatment period to be excluded. The graph display should be generated dynamically based on the user selection of data items. In other words, the database user can select or deselect any data item to be displayed in the
4. Data import and export

Data in the Palm devices can be uploaded into any desktop computer running this database application. It is critical that data from different desktop computers can be merged in any way a database operator desires. In other words, if there are three desktop computers; A, B and C, and each computer contains data, there could be a data overlap. If overlapping data occurs, it means a patient uploaded the same set of data into multiple computers. The database user can merge data from A and B into C, or from B and C into A. During the merge, only unique data will be saved in the database in order to achieve synchronization between the different computers. This is done by exporting tables from one database and importing them into another database.

2.3 Development Tools for Diary

We chose to use Satellite Forms as the development environment for our Palm application since it can integrate with Microsoft Access for data exchange. Microsoft Access is a very convenient database engine for a relatively small database such as ours. It is less expensive and provides very good compatibility with other Windows software. Access 2000 can work with Satellite Forms Active X and Microsoft Graph 2000 Active X for data exchange and visualization. Access 2000 is also shipped with the latest ADO engine for data connectivity.
2.4 Diary Information System Data Flow

Figure 2.1 displays the data flow between different objects in the system. Although theoretically patient and physician should not be in this diagram, we include them to illustrate how the system is used and to portray their roles in the system. The three components in the rectangles and the conduit represent the work done for this thesis.

![Diagram of Diary system flow chart]

Figure 2.1 Diary system flow chart

2.5 Palm Diary Control Flow

Figure 2.2 describes the logic of the Palm Diary program. The majority of the flow chart is self-explanatory. Notice that Palm Diary always catches any incomplete headache. A user has to finish the record for that headache before he or she can
record any other new headache. This is to maintain data integrity. The Diary program checks if there is any missing days since the beginning of the treatment period. A missing day means a day the patient did not record any information. The Palm Diary program will

![Figure 2.2 Palm diary flow chart](image)

Figure 2.2 Palm diary flow chart

remind the patient of all the missing days until the patient provides missing data or two weeks have elapsed. All the data is stored in two tables in the Palm Diary program with the names “Contact” and “Maindata”.
2.6 Database Schema and Data Flow on the Desktop

Figure 2.3 demonstrates the database schema and Figure 2.4 describes how data processing takes place inside the database. Since the Satellite conduit can’t directly transfer data between the Palm and the Access database, two linked tables are used in the database. They are regular db5 files created by Satellite Forms or exported from Access. Access 2000, unlike Access 97, could not modify data in linked tables\(^{13}\), so we created two temporary Access tables (MainDataCopy and ContactCopy) to copy data from the linked tables and process the data in the temporary tables. Keep in mind that the linked and temporary tables only store data for a single treatment period for one patient, while all the other tables store data of more than one treatment period for all patients. Using temporary tables will save a significant amount of time since the database only needs to process the data in the temporary tables.

By examining the data flow in the desktop database as in Figure 2.4, we can see there is a hierarchy for tables in this database. All the data from the Palm is stored in two master tables. The headache data is then extracted and stored in five tables, four of the tables store information for specific types of headaches, ranging from migraine, tension, other headache and unknown headache. The “ALLHA” table stores general information shared by all four types of headaches. The data in the five tables is not adjusted to a 30 day treatment period. The CORE30 table stores information for all the headaches adjusted to a treatment period of 30 days. The information from this table will be
Figure 2.3 Diary database schema on the desktop side
displayed in the tabular or graph summary. Having the processed result in a table can speed up graphical display of data so that displays are almost instantaneous.

2.7 Database Operations in the Desktop Database

This section describes how the database operations are executed in a logical order. Figure 2.5 shows the relationships between different forms in this database application. A user should follow this when switching between different forms.

The functions of the seven forms are discussed below. The main form is a central place for the user to switch between different forms. The standard procedure is:
1. The user starts the database and the splash form is displayed briefly and then
   the main form shows up;
2. On this form, the user can import and export database files, switch to the
   patient information form to edit or search on patient information, switch to
   the summary generation form to generate summaries. The most common
   route is to go to the data upload form to transfer data from the Palm to the
   database;
3. On the data upload form, the user uploads data from the Palm to the database
   and starts the data processing;
4. In the next step, the user goes to the summary generation form to generate
   result summaries. On this form, the user can select headache type and
   exclude treatment periods. The user can choose to generate a tabular or chart
   summary;
5. The tabular summary is a display-only form;
6. The chart summary is very flexible and the user can select any data item as
   indicated by the check boxes.
2.8 Database Export and Import

Table export and import are handled by the two buttons on the MainForm. When one of the buttons is clicked, the user will be asked for confirmation. If the answer is yes, export or import will be executed in the background. In the same desktop folder as the Access database file, there are four other sub folders, namely Files2Export, Files2Import, LinkedFiles and Utilities. They are involved in the data export and import process. The Files2Export is the folder that holds data that has been exported from tables to files. The Files2Import is the folder that holds files to be imported. These files are generated by running the data export function from another computer (for example, computer B) that is running the same database (but a different data
set). The user should manually copy the exported files from computer B into the import folder. When import starts, data in the imported files will be copied into the files in the sub folder LinkedFiles. From there, we run SQL query and load data from the linked files into the Access database tables. Some housekeeping needs to occur here. In the sub folder Utilities, there are a set of tables which contain absolutely no data, but have the same table structures as the files in Files2Export, Files2Import and LinkedFiles. After data import, linked tables have a great amount of data and serve no further purpose. The system then copies the empty files from the sub folder Utilities into the subfolder LinkedFiles in order to purge the data from the linked tables.

2.9 System Verification

Palm Computing provides an excellent tool for development testing. This tool is called the Palm Emulator. An emulator program can take ROM files of different devices and display the corresponding virtual device on the screen. A ROM file is an image of a specific Palm device such as a Palm V or VII which contains the internal structure of a Palm. This allows the emulator to simulate any environment, acting just like the physical device. The Palm computer pictures used in this thesis are all generated by the Palm emulator program. To start an emulator session, click on the program and provide it with a ROM file. This will display a corresponding Palm computer on the screen. A Palm can use a serial port, modem or network connection to do HotSync, while an emulator can only use a network connection to HotSync.
with a desktop computer. The procedure to set up a network HotSync session is described below. It assumes that you have already created an emulator session on a Windows machine.

1. Right click on the emulator and select “Setting” and then “Properties”;
2. Ensure the “Redirect NetLib calls to host TCP/IP” checkbox is checked;
3. Enter a “HotSync User Name” (such as “pose”) and click OK;
4. Open HotSync in the emulator;
5. Select “Modem Sync Prefs” from the HotSync “Options” menu (click on the button on the lower left hand side of the screen to access the menus);
6. Ensure the “Network” option is selected. Click “OK”;
7. Under the “Modem Sync” icon in HotSync the words “Select service” will be displayed. Click on them and select anything from the “Service” popup menu. It does not matter what you select here. Don’t worry about a username or password. Click “Done”;
8. Select “Primary PC Setup” from the HotSync “Options” menu;
9. Enter as much information in this dialog box as you know. If you don’t know anything, simply enter “127.0.0.1” (without quotes!) into the “Primary PC Address” field. Click OK. The IP address 127.0.0.1 stands for the local host. Even if you know your machine’s IP address, you can still use this number and the next time you check your Primary PCs address, it will display the actual IP address for your machine;
10. Right click the HotSync icon in the PC’s system tray and ensure that the “Network” option is checked.

That is all that needs to be done to set up a network HotSync session. Save the session so that you can run it the next time you launch the same emulator session file. When the “Network” option is selected in HotSync, a physical Palm computer can’t do a local HotSync. It will just hang. It doesn’t matter if the “Local” option is checked as long as the “Network” option is checked.

The Palm emulator was used to test the Diary program. By leaving the DataUpload form open in MS Access and running an emulator HotSync, the data from the emulator can be loaded into the Access database. This dramatically reduced testing time. The information system presented in this thesis has been through a complete testing process and its functionality has proven to work as expected.

2.10 Limitations

There is one limitation associated with the Palm Diary program. If a user exits during the middle of recording data, it will result in incomplete data and it is unrecoverable on the Palm platform. This is why data uploaded from the Palm onto the desktop is first copied into two temporary tables (MainDataCopy and ContactCopy). Before we permanently store the data, the data in those two tables are scanned for empty cells. If an empty cell is found, the database user should either fix the incomplete record and process the data if time allows or keep the Palm device and do the data cleaning
later. On the Palm side, this limitation is inherited from Satellite Forms. It is impossible for a Satellite Forms application to check the data integrity. In our case, we have 18 forms for the Palm application. From each form, the user can exit from the program by tapping on one of the four system icons located at the bottom of the screen or pushing any hardware button (except the power button). Tapping on any of these buttons will result in the generation of an incomplete record. To prevent this from happening, the code should be written to check the completeness of a data record when any form closes. Satellite Forms does not support modular and code reuse, so each form has to have this same code. However, this presents a problem because Satellite Forms can’t distinguish between the form close event caused by a user switching from one form to another or the exiting event caused by a user exiting the program. Therefore, it only makes sense to do completeness checking when the user exits from the last form. Unfortunately, this will not work when the user exits from the program before getting to the last form. Because of this limitation, we require our users to finish a record before exiting the program.
Part III  Snapshots of the Diary Information System

This chapter demonstrates the product in snapshots for both the Palm Diary (Figure 3.1.1 to Figure 3.1.22) and the Palm Database (Figure 3.2.1 to Figure 3.2.6) applications. Notice that in some snapshots of Figure 3.1.x, the upper-right corner of the screen shows the form sequence number a user will encounter. These numbers represent the order of forms that are displayed to the user. If multiple forms have the same sequence number, it means these forms are implemented inside one internal form and the components are displayed or hidden according to the program status.

Figure 3.1.1: The letter “D” icon launches the Diary in a Palm

Figure 3.1.2: Patient info configuration form
Figure 3.1.3: The starting form for the Diary program

Figure 3.1.4: Form 2 for a male patient

Figure 3.1.5: Form 2 for a female patient

Figure 3.1.6: Form 3 starts the headache recording process
Figure 3.1.7: Form 4 for a record w/o ending time info

Figure 3.1.8: Form 5 for recording headache detail

Figure 3.1.9: Form 6 for recording headache symptom info

Figure 3.1.10: Form 7 for recording headache medication usage
Figure 3.1.11: Form 8 for Part one of question A

Figure 3.1.12: Form 9 for Part two of question A

Figure 3.1.13: Form 10 for Part one of question B

Figure 3.1.14: Form 11 for Part two of question B
Figure 3.1.15: Form 12 for Part one of question C

Figure 3.1.17: Form 14 for Part one of question D

Figure 3.1.16: Form 13 for Part two of question C

Figure 3.1.18: Form 15 for Part two of question D
Figure 3.1.19: Form 16 for headache record summary

Figure 3.1.20: Form 17 for daily information recorded

Figure 3.1.21: Form 18 to remind the user of missing days

Figure 3.1.22: Form 18 to exit the Program
Figure 3.2.1: Diary database launcher

Figure 3.2.2: Diary database main form
Figure 3.2.3: Form to upload data from the Palm into the database

Figure 3.2.4: Diary database form to create a result chart or table
Figure 3.2.5: Diary database form to display analysis results in a table

Figure 3.2.6: Diary database form to display analysis results in a chart
Part IV  Diary Information System Guide

4.1 General Information

The Palm Diary is developed with Satellite Forms 3.1 Standard Edition. To run Diary, all the 10 pdb and prc files have to be installed on a Palm computer with "Palm Desktop", the PC software provided by Palm Computing when a new Palm is purchased. In addition, Visual Basic 6.0 and Access 2000 should be preinstalled to run the database application.

4.2 Platform Requirements

The Diary information system can be used with any PC running Windows 95/98/NT/2000 (A minimum of memory 128MB, CPU 266MHz and Disk 6.4GB is recommended). If a Visor is used, the connection to the PC is faster using a USB port. However, “Palm Desktop” software only works with one model, either a serial or USB connection. Palm Diary application has been primarily tested on a Palm IIIx, but should work with any model of Palm computers, including HandSpring Visor.

When installing the Diary database on a PC, a user should copy the database file “Diary.mdb” and the four sub folders (Files2Import, Files2Export, LinkedFiles and Utilities) into a folder called “C:\Diary”. This is a mandatory requirement to run the database. In addition, Visual Basic 6.0 and Satellite Forms Standard Edition 3.0 have to be installed to provide the ActiveX and COM units needed for this system.
4.3 Palm Diary Installation

When starting with a fresh Palm computer (without the Diary program), omit step 1 below.

1. Remove all the Diary related programs in the Palm device, including "Diary", "Sat.Forms" and "SF-SE_SysHeap", if they exist. To do so, click on the "Menu" icon located in the left bottom corner of the screen. A drop down list with multiple options will appear. Click on "Delete" and a list of programs will show. Select the program you want to delete and then click the "Delete" button located below the program list. Answer “Yes” to the confirmation message.

2. Start "Palm Desktop" on the desktop PC. When it is running, find the "User" list in the upper right corner of "Palm Desktop" and select the user name for the Palm computer you want to install Diary programs upon.

3. Click the "Install" button in the "Palm Desktop" window and a small window for installation will show up. Click the "Add" button and select all ten files from the folder "C:\Diary\PalmDiary" by clicking the first and the last files while holding down the Shift key. This will highlight all the files. Click on the "Open" button. Dismiss the prompt by clicking “Ok” to close it.

4. Click “Done” in the installation window will spawn a message that says “the selected files will be installed into this user’s Palm computer during the next synchronization.”
5. Put the user’s Palm into the cradle and push the HotSync button on the cradle to start the synchronization. Wait for HotSync to finish. This should take less than one minute.

6. Remove the Palm device from the cradle and check to see if Diary is installed. You should be able to see a letter "D" icon on the screen. It is the launcher for the Palm Diary program.

7. Click on the “D” icon will launch the Diary program. Be sure to fill out the patient information section. Patient information should exist in the desktop database “Diary Data Management System”, an Access database application.

8. Click on the “Done” button and the screen will flash once. Click the home icon to exit the Diary program and return to the main Palm screen.

9. Before giving the Palm to a patient, check the battery and change it if necessary.

4.4 Data Upload

Palm Diary data is stored in two db5 files and can be uploaded into MS Access by running the Diary Database Application by following the procedure below:

1. Start MS Access and open the database C:\Diary\Diary.mdb;

2. Click on "Data Upload" to upload data from a Palm computer;

3. When the Data Upload form comes up, put the Palm computer into the cradle firmly and press the HotSync button on the cradle;
4. After a few moments, you should hear a beeping sound from the Palm. This indicates that HotSync is finished. The current patient’s personal information should be displayed on the data upload form;

5. Be sure to click on the "Process Data" button on the Data Upload form before you leave this form. You will be prompted to check the raw data from the Palm computer for incomplete records (empty cells in the two tables). Open the tables "CONTACT_COPY" and "MAINDATA_COPY" in data sheet mode to check for empty cells. If there are any blank cells, fill them with the default values provided. Blank cells will block the data processing if they are not handled.

4.5 User Guide

1. Patient information, including patient number, treatment period number and patient gender, should be carefully recorded into the Palm computer by the physician who conducts this study. Notice that the pretreatment period is named period zero.

2. Never change the system date in the "Prefs" folder of the Palm computer.

3. If interrupted while using Diary, a user should simply put it down and come back later. Since Palms turn themselves off after 1-3 minutes, the Palm may be powered down. By pushing the green power button the user should be returned to the last Diary screen they were viewing.
4. Never tap the four icons at the bottom of the screen or push any hardware button on the Palm computer while using Diary. Doing so will result in an incomplete record.

5. Diary will not force a user to change any error if the user chooses not to do so when an error message comes up. It is the user’s responsibility to make the corrections.

6. When the battery (indicated by the horizontal bar on the top of the Palm screen) is low or a battery low warning is issued, change the battery with a pair of AAA battery as soon as possible. The batteries should be changed without being interrupted. If a Palm runs out of battery or changing the battery takes more than one minute, data in the Diary application will be lost.

7. Never delete any program installed by a physician. If a user chooses to be a “power user”, he or she may install new programs onto the Palm, however, they still should not delete the Diary programs. We don't encourage users to become power users because they may accidentally delete the Diary program. If a user needs to delete non-Diary programs, she should contact her physician for assistance.
Part V  Conclusions and Future Work

This thesis presented a mobile information system for headache research. The Palm application "Diary" can be taken with patients to record their headache information. The mobility provides a convenient way for patients to actively participate in their headache management and improve the quality of headache research. It helps patients manage their headaches by providing feedback instantaneously on the Palm computer with summary tables and messages. Compared with traditional paper-based methods, it significantly reduces the amount of work needed to process the data and increases the chances of obtaining correct information. Coupled with the desktop Diary database system, a patient can get very precise evaluations on how he or she is managing their headaches. This is a very innovative approach to clinical research and we believe it will be widely accepted as a new standard in the near future.

We need to improve the "Diary" application on the Palm computer to avoid occurrence of incomplete records. Although this is a rare situation under normal use and has been solved on the desktop database application, it should be handled on the Palm platform. This implies that we need to use different development software, such as "Code Warrior", to program the "Diary" application on the Palm. On the database side, we could use a more powerful database engine, such as SQL server or Oracle, to achieve scalability and reliability. Looking into the future, wireless devices are the ideal solution for this type of research because of the availability of
instant feedback for patients without requiring them to come in for an office visit. The next step is to provide online access to patients with security and privacy in mind.
Part VI  Reference


8. “Pendragon Forms 3.1, design and build your own mobile database forms”, Pendragon Software Corporation, June 5th, 2000, URL: http://www.pendragon-software.com


Global Variables:

- `Dim Patient` 'patient number
- `Dim Phase` 'treatment period
- `Dim Gender` '1 as female and 0 as male
- `Dim Missing` 'a flag to indicate if a patient forgot to record info for a past day
- `Dim Cdate` 'date for the current record
- `Dim Cday` 'day for the current record
- `Dim NE` 'if a headache was recorded as not ended, NE = 1, otherwise 0
- `Dim Back` 'if switching from B04 back to B03, Back = 1, default = 0

Configuration Form:

```
OnClick Button_Done
    If Edit_Patient < 1000 Then
        MsgBox("Invalid patient number.")
        Exit
    Else
        Forms("A02_Info").Refresh
EndIf
```
Form 1:

OnClick Button_Start:

Dim Date
Dim Bonus
Dim Ref1

Forms("A01_Start").MoveLast

If Tables("Contact").RecordValid = FALSE Then
    Tables("Contact").CreateRecord
    Tables("Contact").MoveLast
    Tables("Contact").Fields("DATE") = SysDateToDate(GetSysDate)
    Tables("Contact").Fields("REF") = 1
    Tables("Contact").Fields("CONTIMES") = 1
    Tables("Contact").Fields("HADAY") = 0
    Tables("Contact").Fields("BONUS") = 1
    Forms("A02_Info").Show
Else
    Patient = Tables("Contact").Fields("PATIENT")
    Phase = Tables("Contact").Fields("PHASE")
    Gender = Tables("Contact").Fields("GENDER")
    Bonus = Tables("Contact").Fields("BONUS")
    Ref1 = Tables("Contact").Fields("REF")
    Date = DateToSysDate(Tables("Contact").Fields("DATE"))

    ' this should not happen if the patient does not change the system date
    If GetSysDate < Date Then
        MsgBox("System date error, please contact your physician. Exit!")
        Forms("B15_End").Show
    EndIf
' this happens if a patient forgot to record info for one or more days

While GetSysDate > Date

    Tables("Contact").CreateRecord
    Tables("Contact").MoveLast
    Tables("Contact").Fields("PATIENT") = Patient
    Tables("Contact").Fields("PHASE") = Phase
    Tables("Contact").Fields("GENDER") = Gender
    Tables("Contact").Fields("HADAY") = 0
    Tables("Contact").Fields("REF") = Ref1 + 1
    Date = Date + 1
    Tables("Contact").Fields("DATE") = SysDateToDate(Date)
    Tables("Contact").Fields("CONTIMES") = 0

    If Date = GetSysDate Then
        Tables("Contact").Fields("BONUS") = Bonus + 1
        If Bonus Mod 5 = 4 Then
            MsgBox("You have earned " & Bonus*10+10 & " points.")
        EndIf
    Else
        Tables("Contact").Fields("BONUS") = 0
        Bonus = 0
    EndIf

    Ref1 = Ref1 + 1

    Wend

    Tables("Contact").Fields("CONTIMES") =
        Tables("Contact").Fields("CONTIMES") + 1

    Forms("A03_Daily").MoveLast
    Forms("A03_Daily").Show

EndIf
Form 2:

AfterOpen

If Gender = 0 Then
    Text_B1.Visible = False
    Text_B2.Visible = False
    Radio_BYes.Visible = False
    Radio_BNo.Visible = False

EndIf

If Missing = FALSE Then
    Edit_Day = "today,"
else
    Edit_Day = Cday
EndIf

OnClick Button_Next

Tables("MainData").MoveLast
If Tables("MainData").RecordValid = TRUE And
Tables("MainData").Fields("ENDED") = 2 Then
    NE = 1
    Forms("B02_Ne").Show
ElseIf Tables("Contact").Fields("HADAY") = 1 And _
    DateToSysDate(Tables("Contact").Fields("DATE")) <> GetSysDate Then
    NE = 0
    Forms("B15_End").Show
Else
    NE = 0
    Forms("B01_Hd").Show
EndIf

**BeforeClose**

Forms("A03_Daily").Refresh

Form 3:

**AfterOpen**

If Missing = TRUE Then

Text_Today1.Visible = FALSE
Text_Today2.Visible = FALSE
Text_P1.Visible = TRUE
Text_P2.Visible = TRUE
Edit_PDate.Visible = TRUE
Edit_PDay.Visible = TRUE
Edit_PDate = Cdate
Edit_PDay = Cday
Tables("Contact").Fields("MENSTR") = 0

EndIf

**OnClick Button_Back**

Forms("A03_Daily").Show

**OnClick Radio_No**

Forms("B14_Sum").Show

**OnClick Radio_Yes**

Forms("B03_Start").Show
Form 4:

AfterOpen
Dim day
Forms("B02_Ne").MoveLast
day = DateToSysDate(Tables("MainData").Fields("START_DATE")) Mod 7
If day = 0 Then
    Edit_Day = "Friday,"
ElseIf day = 1 Then
    Edit_Day = "Saturday,"
ElseIf day = 2 Then
    Edit_Day = "Sunday,"
ElseIf day = 3 Then
    Edit_Day = "Monday,"
ElseIf day = 4 Then
    Edit_Day = "Tuesday,"
ElseIf day = 5 Then
    Edit_Day = "Wednesday,"
ElseIf day = 6 Then
    Edit_Day = "Thursday,"
EndIf

OnClick Button_Back
Forms("A03_Daily").Show

OnClick Radio_No
Forms("B14_Sum").Show

OnClick Radio_Yes
Forms("B03_Start").Show
Form 5:

AfterOpen

Dim Ref2

Forms("B03_Start").MoveLast

If Back <> 1 Then
    Back = 0
EndIf

If Tables("MainData").RecordValid = FALSE And Back = 0 Then
    Tables("MainData").CreateRecord
    Forms("B03_Start").MoveLast
    Tables("MainData").Fields("REF") = 1
ElseIf NE = 0 And Tables("MainData").RecordValid = TRUE And Back = 0 Then
    Ref2 = Tables("MainData").Fields("REF")
    Tables("MainData").CreateRecord
    Forms("B03_Start").MoveLast
    Tables("MainData").Fields("REF") = Ref2 + 1
EndIf

Tables("MainData").Fields("INPUT_TIME") = SysTimeToTime(GetSysTime)
Tables("MainData").Fields("INPUT_DATE") = SysDateToDate(GetSysDate)
Tables("MainData").Fields("PATIENT") = Patient
Tables("MainData").Fields("PHASE") = Phase
Edit_SD.SetFocus

OnClick Button_Cancel

Dim P
If NE = 1 Then
    Forms("B02_Ne").Show
Else
    If Prompt("Are you sure you want to delete the info in this form?") = 1 Then
        P = Tables("MainData").Position
        Tables("MainData").DeleteRecord(P)
        Forms("B01_Hd").Show
    EndIf
EndIf
Back = 0

OnClick Button_ED
If Tables("MainData").Fields("ENDED") <> 1 Then
    MsgBox("Inconsistent response: You answered No to Has this headache ended?")
Else
    Edit_ED.SetFocus
    Edit_ED = Edit_SD
EndIf
Edit_ED.SetFocus
OnClick Button_EDMinus

If Edit_ED = "" Then
    MsgBox("Click on button End Date first!")
Else
    If DateToSysDate(Edit_ED) <= DateToSysDate(Edit_SD) Then
        MsgBox("Error: End Date before Start Date is not allowed.")
    Else
        Edit_ED = SysDateToData(DateToSysDate(Edit_ED)-1)
    EndIf
EndIf
EndIf

OnClick Button_EDPlus
Edit_ED.SetFocus
If Edit_ED = "" Then
    MsgBox("Click on button End Date first!")
Else
    If DateToSysDate(Edit_ED) >= GetSysDate Then
        MsgBox("Error: Date for the future is not allowed.")
    Else
        Edit_ED = SysDateToDate(DateToSysDate(Edit_ED)+1)
    EndIf
EndIf

OnClick Button_ET
If Tables("MainData").Fields("ENDED") <> 1 Then
    MsgBox("Inconsistent response: You answered No to Has this headache ended?")
Else
    Edit_ET.SetFocus
    Edit_ET = SysTimeToTime(TimeToSysTime(Edit_ST)+900)
EndIf

OnClick Button_ETMinus
Edit_ET.SetFocus
If Edit_ET = "" Then
    MsgBox("Click on button End Time first!")
Else
    If DateToSysDate(Edit_SD) > DateToSysDate(Edit_ED) Then
        MsgBox("Error: Headache End Time has to be after Start Time.")
    Else
        Edit_ED = SysDateToDate(DateToSysDate(Edit_ED)+1)
    EndIf
EndIf
ElseIf DateToSysDate(Edit_SD) = DateToSysDate(Edit_ED) And
    TimeToSysTime(Edit_ET) <= TimeToSysTime(Edit_ST)+900 Then
    MsgBox("Error: Headache End Time has to be after Start Time.")
Else
    Edit_ET = SysTimeToTime(TimeToSysTime(Edit_ET)-900)
EndIf
EndIf

**OnClick Button_ETPPlus**

Edit_ET.SetFocus

If Edit_ET = "" Then
    MsgBox("Click on button End Time first!")
Else
    If DateToSysDate(Edit_ED) > GetSysDate Then
        MsgBox("Error: Time for the future is not allowed.")
    ElseIf DateToSysDate(Edit_ED) = GetSysDate And
        TimeToSysTime(Edit_ET) >= GetSysTime Then
        MsgBox("Error: Time for the future is not allowed.")
    Else
        Edit_ET = SysTimeToTime(TimeToSysTime(Edit_ET)+900)
    EndIf
EndIf
EndIf

**OnClick Button_Next**

Dim fool

Forms("B03_Start").Refresh

If Tables("MainData").Fields("HD_TYPE") = "" Then
    MsgBox("Headache type incomplete!")
Exit
EndIf

If Edit_SD = "" Or Edit_ST = "" Then
    MsgBox("Headache start date and/or time incomplete!")
    Exit
EndIf

If Tables("MainData").Fields("ENDED") <> 1 And
Tables("MainData").Fields("ENDED") <> 2 Then
    MsgBox("Headache end information incomplete!")
    Exit
EndIf

If(Tables("MainData").Fields("ENDED") = 1) Then
    If (Edit_ED = "" Or Edit_ET = ") Then
        MsgBox("Headache ending time and/or date incomplete.")
        Exit
    EndIf
EndIf

If DateToSysDate(Edit_SD) = GetSysDate And TimeToSysTime(Edit_ST) >
GetSysTime
Then
    MsgBox("Error: Headache Start Time in the future not allow!")
    Exit
EndIf

If DateToSysDate(Edit_ED) = GetSysDate And TimeToSysTime(Edit_ET) >
GetSysTime

Then

    MsgBox("Error: Headache End Time in the future not allow!")

    Exit
EndIf

Back = 0
Forms("B04_Sym").Show

**OnClick Button_SD**

    Edit_SD.SetFocus

    If Missing = TRUE Then
        Edit_SD = Cdate
    Else
        Edit_SD = SysDateToDate(GetSysDate)
    EndIf

**OnClick Button_SDMinus**

    Edit_SD.SetFocus

    If Edit_SD = "" Then
        MsgBox("Click on button Start Date first!")
    Else
        Edit_SD = SysDateToDate(DateToSysDate(Edit_SD)-1)
    EndIf

**OnClick Button_SDPlus**

    Edit_SD.SetFocus

    If Edit_SD = "" Then
        MsgBox("Click on button Start Date first!")
    Else
        If DateToSysDate(Edit_SD) >= GetSysDate Then
Msgbox("Error: Date for the future is not allowed.")
Else
    Edit_SD = SysDateToDate(DateToSysDate(Edit_SD)+1)
EndIf
EndIf

OnClick Button_ST
    Edit_ST.SetFocus
    Edit_ST = SysTimeToTime(43200)

OnClick Button_STMinus
    Edit_ST.SetFocus
    If Edit_ST = "" Then
        MsgBox("Click on button Start Time first!")
    Else
        Edit_ST = SysTimeToTime(TimeToSysTime(Edit_ST)-900)
    EndIf

OnClick Button_STPlus
    Edit_ST.SetFocus
    If Edit_ST = "" Then
        MsgBox("Click on button Start Time first!")
    Else
        If DateToSysDate(Edit_SD) > GetSysDate Then
            MsgBox("Error: Time for the future is not allowed.")
        ElseIf DateToSysDate(Edit_SD) = GetSysDate And
            TimeToSysTime(Edit_ST) > GetSysTime Then
            MsgBox("Error: Time for the future is not allowed.")
        Else
            Edit_ST = SysTimeToTime(TimeToSysTime(Edit_ST)+900)
        EndIf
    EndIf
EndIf
**OnClick Radio_No**

If Edit_ED <> "" Or Edit_ET <> "" Then

   If Prompt("Are you sure you want to change the headache ending date &
            Time?") = 1

   Then
   Tables("MainData").Fields("ENDED") = 0
   Edit_ET = ""
   Edit_ED = ""
   EndIf

Else
   Tables("MainData").Fields("ENDED") = 2  ' double check
   Edit_ET = ""
   Edit_ED = ""
   EndIf

**OnClick Radio_Yes**

Tables("MainData").Fields("ENDED") = 1

---

**Form 6:**

**BeforeClose**
Forms("B04_Sym").Refresh

**OnClick Button_Back**
Back = 1
Forms("B03_Start").Show

**OnClick Button_Next**
If Tables("MainData").Fields("MED") = 1 Then
   Forms("B05_Med").Show
Else

Forms("B06_A").Show
EndIf

OnClick DropList_HdSev
If DropList_HdSev = "None" Then
    MsgBox("Headache time info on previous form should be based on associated symptoms below.")
EndIf

OnClick Radio_No
Tables("MainData").Fields("MED") = 0
Tables("MainData").Fields("ANALGESIC") = 0
Tables("MainData").Fields("IMIT_PILL") = 0
Tables("MainData").Fields("IMIT_INJ") = 0
Tables("MainData").Fields("IMIT_SPRAY") = 0
Tables("MainData").Fields("MAXT_PILL") = 0
Tables("MainData").Fields("MAXT_MELT") = 0
Tables("MainData").Fields("REGLAN") = 0
Tables("MainData").Fields("RESCUE") = 0

OnClick Radio_Yes
Tables("MainData").Fields("MED") = 1

Form 7:

Before Close
    Forms("B05_Med").Refresh

OnClick Button_Back
    Forms("B04_Sym").Show
**OnClick Button Next**
If (EditS + EditIP + EditIS + EditII + EditMP + EditMM + EditA + EditR) < 1
Then
   MsgBox("Medication info incomplete!")
Else
   Forms("B06_A").Show
End If

**OnClick ButtonA**
EditA.SetFocus
EditA = 1

**OnClick ButtonAM**
EditA.SetFocus
If EditA = 0 Then
   MsgBox("Negative number not accepted")
Else
   EditA = EditA - 1
End If

**OnClick ButtonAP**
EditA.SetFocus
EditA = EditA + 1

**OnClick ButtonII**
EditII.SetFocus
EditII = 1

**OnClick ButtonIIM**
EditII.SetFocus
If EditII = 0 Then
   MsgBox("Negative number not accepted")
Else
    EditII = EditII - 1
EndIf

OnChange ButtonIIP
    EditII.SetFocus
    EditII = EditII + 1

OnChange ButtonIP
    EditIP.SetFocus
    EditIP = 1

OnChange ButtonIPM
    EditIP.SetFocus
    If EditIP = 0 Then
    MsgBox("Negative number not accepted")
Else
    EditIP = EditIP - 1
EndIf

OnChange ButtonIPP
    EditIP.SetFocus
    EditIP = EditIP + 1

OnChange ButtonIS
    EditIS.SetFocus
    EditIS = 1

OnChange ButtonISM
    EditIS.SetFocus
    If EditIS = 0 Then
MsgBox("Negative number not accepted")

Else
    EditIS = EditIS - 1
EndIf

OnClick ButtonISP
    EditIS.SetFocus
    EditIS = EditIS + 1

OnClick ButtonMH
    EditMP.SetFocus
    EditMP = 1

OnClick ButtonMHM
    EditMP.SetFocus
    If EditMP = 0 Then
        MsgBox("Negative number not accepted")
    Else
        EditMP = EditMP - 1
    EndIf

OnClick ButtonMHP
    EditMP.SetFocus
    EditMP = EditMP + 1

OnClick ButtonML
    EditMM.SetFocus
    EditMM = 1

OnClick ButtonMLM
    EditMM.SetFocus
If EditMM = 0 Then
    MsgBox("Negative number not accepted")
Else
    EditMM = EditMM - 1
EndIf

OnClick ButtonMLP
    EditMM.SetFocus
    EditMM = EditMM + 1

OnClick ButtonR
    EditR.SetFocus
    EditR = 1

OnClick ButtonRM
    EditR.SetFocus
    If EditR = 0 Then
        MsgBox("Negative number not accepted")
    Else
        EditR = EditR - 1
    EndIf

OnClick ButtonRP
    EditR.SetFocus
    EditR = EditR + 1

OnClick ButtonS
    EditS.SetFocus
    EditS = 1

OnClick ButtonSM
    EditS.SetFocus
If EditS = 0 Then
  MsgBox("Negative number not accepted")
Else
  EditS = EditS - 1
EndIf

OnClick ButtonSP
  EditS.SetFocus
  EditS = EditS + 1

Form 8:

AfterLoad
  If Tables("MainData").Fields("ANALGESIC") < 1 Then
    Tables("MainData").Fields("ANALGESIC") = 0
  EndIf
  If Tables("MainData").Fields("IMIT_PILL") < 1 Then
    Tables("MainData").Fields("IMIT_PILL") = 0
  EndIf
  If Tables("MainData").Fields("IMIT_INJ") < 1 Then
    Tables("MainData").Fields("IMIT_INJ") = 0
  EndIf
  If Tables("MainData").Fields("IMIT_SPRAY") < 1 Then
    Tables("MainData").Fields("IMIT_SPRAY") = 0
  EndIf
  If Tables("MainData").Fields("MAXT_PILL") < 1 Then
    Tables("MainData").Fields("MAXT_PILL") = 0
  EndIf
  If Tables("MainData").Fields("MAXT_MELT") < 1 Then
    Tables("MainData").Fields("MAXT_MELT") = 0
  EndIf
If Tables("MainData").Fields("REGLAN") < 1 Then
    Tables("MainData").Fields("REGLAN") = 0
EndIf
If Tables("MainData").Fields("RESCUE") < 1 Then
    Tables("MainData").Fields("RESCUE") = 0
EndIf
Tables("MainData").Fields("TRIPTAN") =
    Tables("MainData").Fields("MAXT_PILL") +
    Tables("MainData").Fields("MAXT_MELT") +
    Tables("MainData").Fields("IMIT_PILL") +
    Tables("MainData").Fields("IMIT_SPRAY") +
    Tables("MainData").Fields("IMIT_INJ")
If Tables("MainData").Fields("ENDED") = 2 Then
    Tables("MainData").Fields("DAYS") = GetSysDate -
        DateToSysDate(Tables("MainData").Fields("START_DATE"))
        + 1
Else
    Tables("MainData").Fields("DAYS") =
        DateToSysDate(Tables("MainData").Fields("END_DATE")) -
        DateToSysDate(Tables("MainData").Fields("START_DATE"))
        + 1
EndIf

BeforeClose
Forms("B06_A").Refresh

OnClick Button_Back
If Tables("MainData").Fields("MED") = 1 Then
    Forms("B05_Med").Show
Else
    Forms("B04_Sym").Show
EndIf

**OnClick ButtonNext**
If Tables("MainData").Fields("EFFECT_A") = 1 Then
    Forms("B07_A").Show
Else
    Forms("B08_B").Show
EndIf

**OnClick Radio No**
Tables("MainData").Fields("EFFECT_A") = 0
Tables("MainData").Fields("MISS_A") = 0
Tables("MainData").Fields("DONE_A") = 0

**OnClick Radio Yes**
Tables("MainData").Fields("EFFECT_A") = 1

**Form 9:**

**AfterOpen**
  EditOne.SetFocus

**BeforeClose**
  Forms("B07_A").Refresh

**OnClick Button_Back**
If (EditOne + EditTwo) = 0 Then
    MsgBox("Warning: Impact info not recorded!")
EndIf
Forms("B06_A").Show
OnClick Button_Next
If (EditOne + EditTwo) = 0 Then
    MsgBox("Warning: Impact info not recorded!")
Else
    Forms("B08_B").Show
EndIf

OnClick ButtonOne
EditOne.SetFocus
EditOne = 0

OnClick ButtonOneM
EditOne.SetFocus
If EditOne = 0 Then
    MsgBox("Negative number not accepted.")
Else
    EditOne = EditOne - 0.5
EndIf

OnClick ButtonOneP
EditOne.SetFocus
EditOne = EditOne + 0.5

OnClick ButtonTwo
EditTwo.SetFocus
EditTwo = 0

OnClick ButtonTwoM
EditTwo.SetFocus
If EditTwo = 0 Then
    MsgBox("Negative number not accepted.")
Else
EditTwo = EditTwo - 0.5
EndIf

OnClick ButtonTwoP
EditTwo.SetFocus
EditTwo = EditTwo + 0.5

Form 10:

Before Close
Forms("B08_B").Refresh

OnClick Button_Back
If Tables("MainData").Fields("EFFECT_A") = 1 Then
   Forms("B07_A").Show
Else
   Forms("B06_A").Show
EndIf

OnClick Button_Next
If Tables("MainData").Fields("EFFECT_B") = 1 Then
   Forms("B09_B").Show
Else
   Forms("B10_C").Show
EndIf

OnClick Radio_No
Tables("MainData").Fields("EFFECT_B") = 0
Tables("MainData").Fields("MISS_B")=0
Tables("MainData").Fields("DONE_B")=0
OnClick Radio_Yes
  Tables("MainData").Fields("EFFECT_B") = 1

Form 11:

AfterOpen
  EditOne.SetFocus

BeforeClose
  Forms("B09_B").Refresh

OnClick Button_Back
  If (EditOne + EditTwo) = 0 Then
    MsgBox("Warning: Impact info not recorded!")
  EndIf
  Forms("B08_B").Show

OnClick Button_Next
  If (EditOne + EditTwo) = 0 Then
    MsgBox("Warning: Impact info not recorded!")
  Else
    Forms("B10_C").Show
  EndIf

OnClick ButtonOne
  EditOne.SetFocus
  EditOne = 0

OnClick ButtonOneM
  EditOne.SetFocus
  If EditOne = 0 Then
    MsgBox("Negative number not accepted.")
Else
    EditOne = EditOne - 0.5
EndIf

OnClick ButtonOneP
    EditOne.SetFocus
    EditOne = EditOne + 0.5

OnClick ButtonTwo
    EditTwo.SetFocus
    EditTwo = 0

OnClick ButtonTwoM
    EditTwo.SetFocus
    If EditTwo = 0 Then
        MsgBox("Negative number not accepted.")
    Else
        EditTwo = EditTwo - 0.5
    EndIf

OnClick ButtonTwoP
    EditTwo.SetFocus
    EditTwo = EditTwo + 0.5
Form 12:

**BeforeClose**
Forms("B10_C").Refresh

**OnClick Button_Back**
If Tables("MainData").Fields("EFFECT_B") = 1 Then
    Forms("B09_B").Show
Else
    Forms("B08_B").Show
EndIf

**OnClick Button_Next**
If Tables("MainData").Fields("EFFECT_C") = 1 Then
    Forms("B11_C").Show
Else
    Forms("B12_D").Show
EndIf

**OnClick Radio_No**
Tables("MainData").Fields("EFFECT_C") = 0
Tables("MainData").Fields("MISS_C") = 0
Tables("MainData").Fields("DONE_C") = 0

**OnClick Radio_Yes**
Tables("MainData").Fields("EFFECT_C") = 1
Form 13:

AfterOpen
   EditOne.SetFocus

BeforeClose
   Forms("B11_C").Refresh

OnClick Button_Back
   If (EditOne + EditTwo) = 0 Then
      MsgBox("Warning: Impact info not recorded!")
   EndIf
   Forms("B10_C").Show

OnClick Button_Next
   If (EditOne + EditTwo) = 0 Then
      MsgBox("Warning: Impact info not recorded!")
   Else
      Forms("B12_D").Show
   EndIf

OnClick ButtonOne
   EditOne.SetFocus
   EditOne = 0

OnClick ButtonOneM
   EditOne.SetFocus
   If EditOne = 0 Then
      MsgBox("Negative number not accepted.")
   Else
      EditOne = EditOne - 0.5
   EndIf
**OnClick ButtonOneP**
EditOne.SetFocus
EditOne = EditOne + 0.5

**OnClick ButtonTwo**
EditTwo.SetFocus
EditTwo = 0

**OnClick ButtonTwoM**
EditTwo.SetFocus
If EditTwo = 0 Then
    MsgBox("Negative number not accepted.")
Else
    EditTwo = EditTwo - 0.5
End If

**OnClick ButtonTwoP**
EditTwo.SetFocus
EditTwo = EditTwo + 0.5

**Form 14:**

**BeforeClose**
Forms("B12_D").Refresh

**OnClick Button_Back**
If Tables("MainData").Fields("EFFECT_C") = 1 Then
    Forms("B11_C").Show
Else
    Forms("B10_C").Show
End If
**OnClick Button_Next**
If Tables("MainData").Fields("EFFECT_D") = 1 Then
  Forms("B13_D").Show
Else
  Forms("B14_Sum").Show
EndIf

**OnClick Radio_No**
Tables("MainData").Fields("EFFECT_D") = 0
Tables("MainData").Fields("MISS_D")=0

**OnClick Radio_Yes**
Tables("MainData").Fields("EFFECT_D") = 1

**Form 15:**

**BeforeClose**
Forms("B13_D").Refresh

**OnClick Button_Back**
If EditOne = 0 Then
  MsgBox("Warning: Impact info not recorded!")
EndIf
Forms("B12_D").Show

**OnClick Button_Next**
If EditOne = 0 Then
  MsgBox("Warning: Impact info not recorded!")
Else
  Forms("B14_Sum").Show
EndIf
**OnClick ButtonOne**

`EditOne = 0`

**OnClick ButtonOneM**

If `EditOne = 0` Then

    `MsgBox("Negative number not accepted.")`

Else

    `EditOne = EditOne - 0.5`

EndIf

`EditOne = EditOne + 0.5`

**OnClick ButtonOneP**

`EditOne = EditOne + 0.5`

---

**Form 16:**

**AfterOpen**

Dim SDate

Tables("Contact").MoveFirst

While Tables("Contact").RecordValid = TRUE

    SDate = DateToSysDate(Tables("Contact").Fields("DATE"))

    If SDate <= DateToSysDate(Tables("MainData").Fields("END_DATE"))

    And

    _ SDate >=

 DateToSysDate(Tables("MainData").Fields("START_DATE"))

    Then

        Tables("Contact").Fields("HADAY") = 1

    EndIf

    If SDate = DateToSysDate(Tables("MainData").Fields("START_DATE"))

    Then

        Tables("Contact").Fields("HADAY") = 1
EndIf
Tables("Contact").MoveNext
Wend

OnClick ButtonNext
Forms("B14_Sum2").Show

Form 17:

OnClick ButtonBack
Forms("B14_Sum").Show

OnClick ButtonNext
Forms("B15_End").Show

Form 18:

AfterOpen
Dim i
Dim Window
Dim temp
Dim Records
' Do not execute the following if there is a record with unended headache
If Tables("MainData").Fields("ENDED") = 1 Or
Tables("MainData").RecordValid = FALSE Then

Records = 0
Tables("Contact").MoveFirst
While Tables("Contact").RecordValid = TRUE
    Records = Records + 1
    Tables("Contact").MoveNext
Wend

If Records > 15 Then
   Window = 15
Else
   Window = Records
EndIf

Forms("B15_End").MoveLast
For i = 1 to Window - 1
   Forms("B15_End").MovePrevious
Next i

For i = 1 to Window - 1
   If Tables("Contact").Fields("CONTIMES") = 0 Then
      Missing = TRUE
      Text_A.Visible = TRUE
      Text_B.Visible = TRUE
      Cdate = Tables("Contact").Fields("DATE")
      temp = DateToSysDate(Cdate) Mod 7
      If temp = 0 Then
         Cday = "Friday,"
      ElseIf   temp = 1 Then
         Cday = "Saturday,"
      ElseIf   temp = 2 Then
         Cday = "Sunday,"
      ElseIf   temp = 3 Then
         Cday = "Monday,"
      ElseIf   temp = 4 Then
Cday = "Tuesday,"
ElseIf temp = 5 Then
    Cday = "Wednesday,"
ElseIf temp = 6 Then
    Cday = "Thursday,"
EndIf
If Prompt("Would you like to record any information for " & Cday & 
    Cdate & "?") = 1 Then
    Tables("Contact").Fields("CONTIMES") = 1
    Forms("A03_Daily").Show
    Exit
EndIf
EndIf
Forms("B15_End").MoveNext
Next i
EndIf

Text_A.Visible = FALSE
Text_B.Visible = FALSE
Text_Thanks.Visible = TRUE
Text_Bye.Visible = TRUE
Bitmap_Smile.Visible = TRUE
7.2 Diary Database Source Code

1. Module

Option Compare Database
Public gvMode As Integer
  ' 0 : patient mode
  ' 1 : group mode
Public gvHeadache As Integer
  ' 1 : Migraine
  ' 2 : Tension
  ' 3 : Other
  ' 4 : Unknown
  ' 5 : All
Public gvPatientNum As Integer
Public gvGroupNum As Integer
Public gvTreat(8) As Boolean

2. Launch up Form

Option Compare Database
Option Explicit

Private Sub Form_Open(Cancel As Integer)
  Me.TimerInterval = 2500
End Sub

Private Sub Form_Timer()
  DoCmd.Close acForm, "frmStart"
End Sub
3. Task Switch Form

Option Compare Database

Private Sub ButtonExit_Click()
    If (MsgBox("Are you sure you want to quit the program?", vbYesNo) = vbYes) Then
        DoCmd.Quit
    End If
End Sub

Private Sub ButtonExport_Click()
    Dim vbAnswer
    If (MsgBox("Are you sure you want to export the database?", vbYesNo) = vbNo) Then
        Exit Sub
    Else
        DoCmd.SetWarnings 0
        Set fs = CreateObject("Scripting.FileSystemObject")
        If (fs.FileExists("C:\Diary\Files2Export\Wcontact.dbf")) Then
            Kill "C:\Diary\Files2Export\Wcontact.dbf"
        End If
If (fs.FileExists("C:\Diary\Files2Export\Wmaindat.dbf")) Then
    Kill "C:\Diary\Files2Export\Wmaindat.dbf"
End If

If (fs.FileExists("C:\Diary\Files2Export\Wpatient.dbf")) Then
    Kill "C:\Diary\Files2Export\Wpatient.dbf"
End If

If (fs.FileExists("C:\Diary\Files2Export\Wsecret.dbf")) Then
    Kill "C:\Diary\Files2Export\Wsecret.dbf"
End If

If (fs.FileExists("C:\Diary\Files2Export\Wcore30.dbf")) Then
    Kill "C:\Diary\Files2Export\Wcore30.dbf"
End If

If (fs.FileExists("C:\Diary\Files2Export\Wallha.dbf")) Then
    Kill "C:\Diary\Files2Export\Wallha.dbf"
End If

If (fs.FileExists("C:\Diary\Files2Export\Wmigrain.dbf")) Then
    Kill "C:\Diary\Files2Export\Wmigrain.dbf"
End If

If (fs.FileExists("C:\Diary\Files2Export\Wtension.dbf")) Then
    Kill "C:\Diary\Files2Export\Wtension.dbf"
End If

If (fs.FileExists("C:\Diary\Files2Export\Wother.dbf")) Then

Kill "C:\Diary\Files2Export\Wother.dbf"
End If

If (fs.FileExists("C:\Diary\Files2Export\Wunknown.dbf")) Then
    Kill "C:\Diary\Files2Export\Wunknown.dbf"
End If

' export data from database into utilities files
DoCmd.OpenQuery "ExportContact"
DoCmd.OpenQuery "ExportMaindata"
DoCmd.OpenQuery "ExportPatient"
DoCmd.OpenQuery "ExportSecret"
DoCmd.OpenQuery "ExportCore30"
DoCmd.OpenQuery "ExportAllha"
DoCmd.OpenQuery "ExportMigraine"
DoCmd.OpenQuery "ExportTension"
DoCmd.OpenQuery "ExportUnknown"
DoCmd.OpenQuery "ExportOtherHA"

MsgBox ("Data has been successfully exported!")
End If
End Sub

Private Sub ButtonImport_Click()
    If (MsgBox("Are you sure you want to import the external data?", vbYesNo) = vbNo)
Then
    Exit Sub
Else

DoCmd.SetWarnings 0
' copy source data into link files if they exist
Set fs = CreateObject("Scripting.FileSystemObject")

If (fs.FileExists("C:\Diary\Files2Import\Wcontact.dbf")) Then
    FileCopy "C:\Diary\Files2Import\Wcontact.dbf",
        "C:\Diary\LinkedFiles\Lcontact.dbf"
Else
    MsgBox "Couldn't find file Wcontact.dbf"
    Exit Sub
End If

If (fs.FileExists("C:\Diary\Files2Import\Wmaindat.dbf")) Then
    FileCopy "C:\Diary\Files2Import\Wmaindat.dbf",
        "C:\Diary\LinkedFiles\Lmaindat.dbf"
Else
    MsgBox "Couldn't find file Wmaindat.dbf"
    Exit Sub
End If

If (fs.FileExists("C:\Diary\Files2Import\Wpatient.dbf")) Then
    FileCopy "C:\Diary\Files2Import\Wpatient.dbf",
        "C:\Diary\LinkedFiles\Lpatient.dbf"
Else
    MsgBox "Couldn't find file Wpatient.dbf"
    Exit Sub
End If

If (fs.FileExists("C:\Diary\Files2Import\Wsecret.dbf")) Then
FileCopy "C:\Diary\Files2Import\Wsecret.dbf",
    "C:\Diary\LinkedFiles\Lsecret.dbf"
Else
    MsgBox " Couldn't find file Wsecret.dbf"
    Exit Sub
End If

If (fs.FileExists("C:\Diary\Files2Import\Wcore30.dbf")) Then
    FileCopy "C:\Diary\Files2Import\Wcore30.dbf",
    "C:\Diary\LinkedFiles\Lcore30.dbf"
Else
    MsgBox " Couldn't find file Wcore30.dbf"
    Exit Sub
End If

If (fs.FileExists("C:\Diary\Files2Import\Wallha.dbf")) Then
    FileCopy "C:\Diary\Files2Import\Wallha.dbf",
    "C:\Diary\LinkedFiles\Lallha.dbf"
Else
    MsgBox " Couldn't find file Wallha.dbf"
    Exit Sub
End If

If (fs.FileExists("C:\Diary\Files2Import\Wmigrain.dbf")) Then
    FileCopy "C:\Diary\Files2Import\Wmigrain.dbf",
    "C:\Diary\LinkedFiles\Lmigrain.dbf"
Else
    MsgBox " Couldn't find file Wmigrain.dbf"
    Exit Sub
End If
End If

If (fs.FileExists("C:\Diary\Files2Import\Wtension.dbf")) Then
    FileCopy "C:\Diary\Files2Import\Wtension.dbf",
    "C:\Diary\LinkedFiles\Ltension.dbf"
Else
    MsgBox "Couldn't find file Wtension.dbf"
    Exit Sub
End If

If (fs.FileExists("C:\Diary\Files2Import\Wother.dbf")) Then
    FileCopy "C:\Diary\Files2Import\Wother.dbf",
    "C:\Diary\LinkedFiles\Lother.dbf"
Else
    MsgBox "Couldn't find file Wother.dbf"
    Exit Sub
End If

If (fs.FileExists("C:\Diary\Files2Import\Wunknown.dbf")) Then
    FileCopy "C:\Diary\Files2Import\Wunknown.dbf",
    "C:\Diary\LinkedFiles\Lunknown.dbf"
Else
    MsgBox "Couldn't find file Wunknown.dbf"
    Exit Sub
End If

' import data from link files into database
DoCmd.OpenQuery "ImportContact"
DoCmd.OpenQuery "ImportMaindata"
DoCmd.OpenQuery "ImportPatient"
DoCmd.OpenQuery "ImportSecret"
DoCmd.OpenQuery "ImportCore30"
DoCmd.OpenQuery "ImportAllha"
DoCmd.OpenQuery "ImportMigraine"
DoCmd.OpenQuery "ImportTension"
DoCmd.OpenQuery "ImportUnknown"
DoCmd.OpenQuery "ImportOtherHA"

' clean up data in link files for future useage
FileCopy "C:\Diary\Utilities\Ucontact.dbf", "C:\Diary\LinkedFiles\Lcontact.dbf"
FileCopy "C:\Diary\Utilities\Umaindat.dbf", "C:\Diary\LinkedFiles\Lmaindat.dbf"
FileCopy "C:\Diary\Utilities\Upatient.dbf", "C:\Diary\LinkedFiles\Lpatient.dbf"
FileCopy "C:\Diary\Utilities\Usecret.dbf", "C:\Diary\LinkedFiles\Lsecret.dbf"
FileCopy "C:\Diary\Utilities\Ucore30.dbf", "C:\Diary\LinkedFiles\Lcore30.dbf"
FileCopy "C:\Diary\Utilities\Uallha.dbf", "C:\Diary\LinkedFiles\Lallha.dbf"
FileCopy "C:\Diary\Utilities\Umigrain.dbf", "C:\Diary\LinkedFiles\Lmigrain.dbf"
FileCopy "C:\Diary\Utilities\Utension.dbf", "C:\Diary\LinkedFiles\Ltension.dbf"
FileCopy "C:\Diary\Utilities\Uother.dbf", "C:\Diary\LinkedFiles\Lother.dbf"
FileCopy "C:\Diary\Utilities\Uunknown.dbf", "C:\Diary\LinkedFiles\Lunknown.dbf"

MsgBox ("External data has been successfully imported!")

End If
End Sub
Private Sub ButtonPatient_Click()
    DoCmd.Close acForm, "frmMain"
    DoCmd.OpenForm "frmPatient"
End Sub

Private Sub ButtonSummary_Click()
    DoCmd.Close acForm, "frmMain"
    DoCmd.OpenForm "frmSummary"
End Sub

Private Sub ButtonUpload_Click()
    DoCmd.SetWarnings False
    'C:\Diary\'
    'The following serves to empty the records in 2 linked db5 files.
    Kill "C:\Diary\LinkedFiles\MAINDATA.dbf"
    Kill "C:\Diary\LinkedFiles\CONTACT.dbf"
    FileCopy "C:\Diary\Utilities\MAINDATA.dbf",
    "C:\Diary\LinkedFiles\MAINDATA.dbf"
    FileCopy "C:\Diary\Utilities\CONTACT.dbf",
    "C:\Diary\LinkedFiles\CONTACT.dbf"
    'The following is to empty the records in two access tables
    DoCmd.OpenQuery "EmptyContact_Copy"
    DoCmd.OpenQuery "EmptyMainData_Copy"
    DoCmd.Close acForm, "frmMain"
    DoCmd.OpenForm "frmDataUpload"
End Sub

4. Data Upload Form

Option Compare Database
Option Explicit

Const Status_HotSyncStart = 1
Const Status_HotSyncEnd = 2
Const Status_HotSyncCommandComplete = 3

Dim MainData_Table As String
Dim Contact_Table As String

Dim HotSync_Progress As String
Dim CmdCount As Integer
Dim ProcessControl As Integer

Private Sub Form_Close()
    If ProcessControl = 1 Then
        MsgBox ("Error: You have not processed the data you just uploaded. Upload the
data again and push the Process button.")
        DoCmd.OpenForm "frmDataUpload"
    End If
End Sub

Private Sub Form_Load()
    'Disable Warnings
    ProcessControl = 0
    DoCmd.SetWarnings 0
    SatForms.Enabled = True
    MainData_Table = "LinkedFiles\MAINDATA.dbf"
    Contact_Table = "LinkedFiles\CONTACT.dbf"
HotSync_Progress = "Begin"
End Sub

Private Sub SatForms_HotSyncStatus(ByVal StatusCode As Long, ByVal Param
As Long)
    TxtDeviceNum.SetFocus
    TxtDeviceNum.Text = SatForms.PilotUserName
    If StatusCode = Status_HotSyncEnd Then
        HotSync_Progress = "End" 
        ButtonProcess.SetFocus 
        Exit Sub
    End If
    If StatusCode = Status_HotSyncStart Then
        SatForms.GetTableFromPalmPilot (MainData_Table)
        SatForms.GetTableFromPalmPilot (Contact_Table)
        CmdCount = 2
        HotSync_Progress = "A>B"
    End If
    If StatusCode = Status_HotSyncCommandComplete Then
        CmdCount = CmdCount - 1
        If CmdCount <> 0 Then GoTo CmdCompleteExit
        If HotSync_Progress = "A>B" Then 
            DoCmd.OpenQuery "CopyContact"
            DoCmd.OpenQuery "CopyMainData"
Private Sub ButtonSummary_Click()
    If ProcessControl = 1 Then
        MsgBox("Warning: You have not processed the uploaded data, which will result in data loss.")
        Exit Sub
    End If
End Sub

Private Sub ButtonProcess_Click()
    ' Need to implement 1. check empty cells, 2. if partially process, undo
    Dim i As Integer

    Dim cat As New ADOX.Catalog
    Dim cnn As New ADODB.Connection
    Dim rst As New ADODB.Recordset
    Dim rst0 As New ADODB.Recordset
    Dim rst1 As New ADODB.Recordset
    Dim rst2 As New ADODB.Recordset
Dim rst3 As New ADODB.Recordset

Dim strSQL0 As String
Dim strSQL1 As String
Dim strSQL2 As String
Dim strSQL3 As String

Dim rstContact As New ADODB.Recordset
Dim rstMaindata As New ADODB.Recordset
Dim strContact As String
Dim strMaindata As String

DoCmd.SetWarnings 0

If ProcessControl = 2 Then
    MsgBox ("Data already processed!")
    Exit Sub
End If

If ProcessControl = 0 Then
    MsgBox ("No data to process.")
    Exit Sub
End If

Set cnn = CurrentProject.Connection
Set cat.ActiveConnection = cnn
Set rst.ActiveConnection = cnn

strSQL0 = "Select DISTINCT PATIENT_INFO.FNAME,"
PATIENT_INFO.LNAME," & 
"CONTACT_COPY.PATIENT, CONTACT_COPY.PHASE " & _
"From PATIENT_INFO, CONTACT_COPY Where 
CONTACT_COPY.PATIENT = " & _
"PATIENT_INFO.PATIENT;"

rst0.Open strSQL0, cnn

If rst0.EOF = True Then 
    MsgBox "Patient info not available in database. Check the patient info table and upload data again."
    ProcessControl = 2
    Exit Sub
Else
    TxtName.SetFocus
    TxtName.Text = rst0.Fields(0) & " " & rst0.Fields(1)
    TxtPatient.SetFocus
    TxtPatient.Text = rst0.Fields(2)
    TxtPhase.SetFocus
    TxtPhase.Text = rst0.Fields(3)
End If

' we probably want to do the check here before proceeding
rstContact.ActiveConnection = cnn
rstMaindata.ActiveConnection = cnn

strContact = "Select MENSTR, DAILYMED From CONTACT_COPY;"
strMaindata = "Select * From MAINDATA_COPY;"
rstContact.Open strContact, cnn
rstMaindata.Open strMaindata, cnn

Do Until rstContact.EOF
   For i = 0 To 1
      If rstContact.Fields(i) = -1 Then
         MsgBox "Incomplete record(s) found in table CONTACT_COPY"
         ProcessControl = 2
         Exit Sub
      End If
   Next i
   rstContact.MoveNext
Loop

Do Until rstMaindata.EOF
   For i = 0 To 29
      If rstMaindata.Fields(i) = -1 Then
         MsgBox "Incomplete records(s) found in table MAINDATA_COPY"
         ProcessControl = 2
         Exit Sub
      End If
   Next i
   rstMaindata.MoveNext
Loop

' Now it is safe to proceed the data

DoCmd.OpenQuery "CopyMainDataStep2"
DoCmd.OpenQuery "AppendMainData"
DoCmd.OpenQuery "AppendContact"

strSQL1 = "SELECT DISTINCT CONTACT_COPY.PATIENT, CONTACT_COPY.PHASE, " & "Max(CONTACT_COPY.REF), Sum(CONTACT_COPY.DAILYMED), " & 
"Sum(CONTACT_COPY.HADAY), Max(CONTACT_COPY.BONUS) FROM CONTACT_COPY " & 
"GROUP BY CONTACT_COPY.PATIENT, CONTACT_COPY.PHASE;"

strSQL2 = "SELECT Count(CONTACT_COPY.CONTIMES)FROM CONTACT_COPY " & 
"WHERE CONTACT_COPY.CONTIMES > 0 GROUP BY CONTACT_COPY.PATIENT, CONTACT_COPY.PHASE;"

strSQL3 = "SELECT Sum(MAINDATA_COPY.SDAYS), Max(MAINDATA_COPY.REF)," & 
" Avg(MAINDATA_COPY.MAX_SEV)," & _
" Sum(MAINDATA_COPY.DEHOURS), Sum(MAINDATA_COPY.DHOURS)," & _
" Sum(MAINDATA_COPY.MISS_D), Sum(MAINDATA_COPY.ANALGESIC)," & _
" Sum(MAINDATA_COPY.IMIT_PILL), Sum(MAINDATA_COPY.IMIT_SPRAY)," & _
" Sum(MAINDATA_COPY.IMIT_INJ), Sum(MAINDATA_COPY.MAXT_PILL)," & _
" Sum(MAINDATA_COPY.MAXT_MELT), Sum(MAINDATA_COPY.REGLAN)," & _
" Sum(MAINDATA_COPY.RESCUE) FROM MAINDATA_COPY" & _
" GROUP BY MAINDATA_COPY.PATIENT, 
MAINDATA_COPY.PHASE; "

rst1.Open strSQL1, cnn 
rst2.Open strSQL2, cnn 
rst3.Open strSQL3, cnn 

rst.Open "ALLHA", , adOpenKeyset, adLockOptimistic, adCmdTable 
If rst1.EOF = False And rst2.EOF = False And rst3.EOF = False Then 
   With rst 
      .AddNew 
         .Fields(0) = rst1.Fields(0) 
         .Fields(1) = rst1.Fields(1) 
         .Fields(2) = rst1.Fields(2) 
         .Fields(3) = rst1.Fields(2) - rst2.Fields(0) 
         .Fields(4) = rst1.Fields(3) 
         .Fields(5) = rst1.Fields(4) 
         .Fields(6) = rst3.Fields(0) 
         .Fields(7) = rst3.Fields(1) 'ha count 
         .Fields(8) = rst1.Fields(5) 'ha days 
         .Fields(9) = rst3.Fields(2) 
         .Fields(10) = rst3.Fields(3) 
         .Fields(11) = rst3.Fields(4) 
         .Fields(12) = rst3.Fields(5) 
         .Fields(13) = rst3.Fields(6) 
         .Fields(14) = rst3.Fields(7) 
         .Fields(15) = rst3.Fields(8) 
         .Fields(16) = rst3.Fields(9) 
         .Fields(17) = rst3.Fields(10)
.Fields(18) = rst3.Fields(11)
.Fields(19) = rst3.Fields(12)
.Fields(20) = rst3.Fields(13)
.Update
End With
Else
    MsgBox ("Error in selected data recordset for table ALLHA.")
    Exit Sub
End If
End Sub

DoCmd.OpenQuery "AppendMigraine"
DoCmd.OpenQuery "AppendTension"
DoCmd.OpenQuery "AppendOtherHA"
DoCmd.OpenQuery "AppendUnknown"
DoCmd.OpenQuery "AppendCore1"
DoCmd.OpenQuery "AppendCore2"
DoCmd.OpenQuery "AppendCore3"
DoCmd.OpenQuery "AppendCore4"
DoCmd.OpenQuery "AppendCore5"

ProcessControl = 2
End Sub

5. Summary Generation Form

Option Compare Database
Option Explicit
Private Sub AllOPt_MouseDown(Button As Integer, Shift As Integer, X As Single,
Y As Single)
    gvHeadache = 5
End Sub

Private Sub ButtonReport_Click()
    gvTreat(0) = ChkT0.Value
    gvTreat(1) = ChkT1.Value
    gvTreat(2) = ChkT2.Value
    gvTreat(3) = ChkT3.Value
    gvTreat(4) = ChkT4.Value
    gvTreat(5) = ChkT5.Value
    gvTreat(6) = ChkT6.Value
    gvTreat(7) = ChkT7.Value

    If gvPatientNum = 9999 And gvGroupNum = 0 Then
        MsgBox "Patient or Group number not selected"
        Exit Sub
    Else
        If gvMode = False Then
            gvPatientNum = ComboPatient.Value
        Else
            gvGroupNum = ComboGroup.Value
        End If
    End If
End If

DoCmd.Close acForm, "frmSummary"
DoCmd.OpenForm "frmReport"
End Sub

Private Sub ButtonChart_Click()
    gvTreat(0) = ChkT0.Value
    gvTreat(1) = ChkT1.Value
    gvTreat(2) = ChkT2.Value
    gvTreat(3) = ChkT3.Value
    gvTreat(4) = ChkT4.Value
    gvTreat(5) = ChkT5.Value
    gvTreat(6) = ChkT6.Value
    gvTreat(7) = ChkT7.Value

    If gvPatientNum = 9999 And gvGroupNum = 0 Then
        MsgBox "Patient or Group number not selected"
        Exit Sub
    Else
        If gvMode = False Then
            gvPatientNum = ComboPatient.Value
        Else
            gvGroupNum = ComboGroup.Value
        End If
    End If

    DoCmd.Close acForm, "frmSummary"
    DoCmd.OpenForm "frmChart"
End Sub

Private Sub ButtonDone_Click()
    DoCmd.Close acForm, "frmSummary"
    DoCmd.OpenForm "frmMain"
End Sub

Private Sub ComboGroup_Change()
    gvGroupNum = ComboGroup.Value
End Sub

Private Sub ComboPatient_Change()
    gvPatientNum = ComboPatient.Value
End Sub

Private Sub Form_Load()
    Dim i
    ComboPatient.Enabled = True
    ComboGroup.Enabled = False
    gvMode = False  ' default mode = patient mode
    gvPatientNum = ComboPatient.Value
    gvGroupNum = ComboGroup.Value
    gvHeadache = 1

    ChkT0.Value = False
    ChkT1.Value = False
    ChkT2.Value = False
    ChkT3.Value = False
    ChkT4.Value = False
    ChkT5.Value = False
    ChkT6.Value = False
    ChkT7.Value = False
End Sub
Private Sub MigraineOpt_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    gvHeadache = 1
End Sub

Private Sub OptGroup_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    ComboPatient.Enabled = False
    ComboGroup.Enabled = True
    gvMode = True

    ChkT0.Value = False
    ChkT1.Value = False
    ChkT2.Value = False
    ChkT3.Value = False
    ChkT4.Value = False
    ChkT5.Value = False
    ChkT6.Value = False
    ChkT7.Value = False
End Sub

Private Sub OptPatient_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Dim i
    ComboPatient.Enabled = True
    ComboGroup.Enabled = False
    gvMode = False
ChkT0.Value = False
ChkT1.Value = False
ChkT2.Value = False
ChkT3.Value = False
ChkT4.Value = False
ChkT5.Value = False
ChkT6.Value = False
ChkT7.Value = False

End Sub

Private Sub OtherOpt_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
gvHeadache = 3
End Sub

Private Sub TensionOpt_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
gvHeadache = 2
End Sub

Private Sub UnknownOpt_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
gvHeadache = 4
End Sub

6. Summary Table Display Form
Option Compare Database

Private Sub Form_Close()
    DoCmd.OpenForm "frmSummary"
End Sub

Private Sub Form_Load()
    Dim rst As New ADODB.Recordset
    Dim cnn As New ADODB.Connection
    Dim strSQL As String
    Dim i As Integer

    If gvMode = False Then
        Title.Caption = "Treatment Summary for Individual"
        WhatNum.Caption = "Patient Num."
        Number.SetFocus
        Number.Text = gvPatientNum
CORE30.PATIENT = " & _
gvPatientNum & " And CORE30.HATYPE = " & gvHeadache & " ;"
Else
  Title.Caption = "Treatment Summary for Group"
  WhatNum.Caption = "Group Num:"
  Number.SetFocus
  Number.Text = gvGroupNum
  strSQL = "Select Distinct CORE30.PHASE, Avg(CORE30.HATYPE),
                  Avg(CORE30.PATIENT)," & _
                  "Avg(CORE30.TREAT_DAYS)As
TreatDays,Avg(CORE30.MISS_DAYS),
                  " & _ "Avg(CORE30.PMT_DAYS), Avg(CORE30.BONUS),
                  Avg(CORE30.HADAYS), Avg(CORE30.SHADAYS), " & _
                  "Avg(CORE30.HACCOUNT), Avg(CORE30.MAX_AVGSEV),
                  Avg(CORE30.DEHOURS), " & _
                  "Avg(CORE30.DHOURS), Avg(CORE30.LSHOURS),
                  Avg(CORE30.ANALGESIC), " &
                  _ "Avg(CORE30.IMPILL), Avg(CORE30.IMSPRAY),
                  Avg(CORE30.IMINI), " & _
                  "Avg(CORE30.MAXTPILL), Avg(CORE30.MAXTMELT),
                  Avg(CORE30.REGLAN), " &
                  _ "Avg(CORE30.RESCUE) From CORE30, SECRET_TBL " & _
                  "Where CORE30.PATIENT = SECRET_TBL.PATIENT And
                  SECRET_TBL.GROUP = "
                  & gvGroupNum & _ " And CORE30.HATYPE = " & _
                  gvHeadache & " Group by CORE30.PHASE ;"
End If

HAType.SetFocus
If gvHeadache = 1 Then
HAType.Text = "Migraine"
ElseIf gvHeadache = 5 Then
    HAType.Text = "All Headaches"
ElseIf gvHeadache = 2 Then
    HAType.Text = "Tension Headache"
ElseIf gvHeadache = 3 Then
    HAType.Text = "Other Headache"
ElseIf gvHeadache = 4 Then
    HAType.Text = "Unknown Headache"
Else
    MsgBox ("Error in headache type!")
    Exit Sub
End If

Set cnn = CurrentProject.Connection
rst.Open strSQL, cnn

Do Until rst.EOF
    Select Case rst.Fields(0)
        Case 0
            If gvTreat(0) = False Then
                TD0.SetFocus
                TD0.Text = rst.Fields(3)
                MSD0.SetFocus
                MSD0.Text = rst.Fields(4)
                PMTD0.SetFocus
                PMTD0.Text = rst.Fields(5)
                B0.SetFocus
                B0.Text = rst.Fields(6)
IDO.SetFocus
IDO.Text = rst.Fields(7)
SHAD0.SetFocus
SHAD0.Text = rst.Fields(8)
HAC0.SetFocus
HAC0.Text = rst.Fields(9)
MAS0.SetFocus
MAS0.Text = rst.Fields(10)
DEH0.SetFocus
DEH0.Text = rst.Fields(11)
DH0.SetFocus
DH0.Text = rst.Fields(12)
LSH0.SetFocus
LSH0.Text = rst.Fields(13)
SA0.SetFocus
SA0.Text = rst.Fields(14)
IP0.SetFocus
IP0.Text = rst.Fields(15)
IS0.SetFocus
IS0.Text = rst.Fields(16)
II0.SetFocus
II0.Text = rst.Fields(17)
MP0.SetFocus
MP0.Text = rst.Fields(18)
MS0.SetFocus
MS0.Text = rst.Fields(19)
RA0.SetFocus
RA0.Text = rst.Fields(20)
RM0.SetFocus
Case 1
If gvTreat(1) = False Then
    TD1.SetFocus
    TD1.Text = rst.Fields(3)
    MSD1.SetFocus
    MSD1.Text = rst.Fields(4)
    PMTD1.SetFocus
    PMTD1.Text = rst.Fields(5)
    B1.SetFocus
    B1.Text = rst.Fields(6)
    HD1.SetFocus
    HD1.Text = rst.Fields(7)
    SHAD1.SetFocus
    SHAD1.Text = rst.Fields(8)
    HAC1.SetFocus
    HAC1.Text = rst.Fields(9)
    MASI.SetFocus
    MASI.Text = rst.Fields(10)
    DEH1.SetFocus
    DEH1.Text = rst.Fields(11)
    DH1.SetFocus
    DH1.Text = rst.Fields(12)
    LSH1.SetFocus
    LSH1.Text = rst.Fields(13)
    SA1.SetFocus
    SA1.Text = rst.Fields(14)
    IP1.SetFocus
End If
IP1.Text = rst.Fields(15)
IS1.SetFocus
IS1.Text = rst.Fields(16)
II1.SetFocus
II1.Text = rst.Fields(17)
MP1.SetFocus
MP1.Text = rst.Fields(18)
MS1.SetFocus
MS1.Text = rst.Fields(19)
RA1.SetFocus
RA1.Text = rst.Fields(20)
RM1.SetFocus
RM1.Text = rst.Fields(21)
End If

Case 2
If gvTreat(2) = False Then
  TD2.SetFocus
  TD2.Text = rst.Fields(3)
  MSD2.SetFocus
  MSD2.Text = rst.Fields(4)
  PMTD2.SetFocus
  PMTD2.Text = rst.Fields(5)
  B2.SetFocus
  B2.Text = rst.Fields(6)
  HD2.SetFocus
  HD2.Text = rst.Fields(7)
  SHAD2.SetFocus
  SHAD2.Text = rst.Fields(8)
  HAC2.SetFocus
HAC2.Text = rst.Fields(9)
MAS2.SetFocus
MAS2.Text = rst.Fields(10)
DEH2.SetFocus
DEH2.Text = rst.Fields(11)
DH2.SetFocus
DH2.Text = rst.Fields(12)
LSH2.SetFocus
LSH2.Text = rst.Fields(13)
SA2.SetFocus
SA2.Text = rst.Fields(14)
IP2.SetFocus
IP2.Text = rst.Fields(15)
IS2.SetFocus
IS2.Text = rst.Fields(16)
II2.SetFocus
II2.Text = rst.Fields(17)
MP2.SetFocus
MP2.Text = rst.Fields(18)
MS2.SetFocus
MS2.Text = rst.Fields(19)
RA2.SetFocus
RA2.Text = rst.Fields(20)
RM2.SetFocus
RM2.Text = rst.Fields(21)
End If

Case 3
If gvTreat(3) = False Then
TD3.SetFocus
TD3.Text = rst.Fields(3)
MSD3.SetFocus
MSD3.Text = rst.Fields(4)
PMTD3.SetFocus
PMTD3.Text = rst.Fields(5)
B3.SetFocus
B3.Text = rst.Fields(6)
HD3.SetFocus
HD3.Text = rst.Fields(7)
SHAD3.SetFocus
SHAD3.Text = rst.Fields(8)
HAC3.SetFocus
HAC3.Text = rst.Fields(9)
MAS3.SetFocus
MAS3.Text = rst.Fields(10)
DEH3.SetFocus
DEH3.Text = rst.Fields(11)
DH3.SetFocus
DH3.Text = rst.Fields(12)
LSH3.SetFocus
LSH3.Text = rst.Fields(13)
SA3.SetFocus
SA3.Text = rst.Fields(14)
IP3.SetFocus
IP3.Text = rst.Fields(15)
IS3.SetFocus
IS3.Text = rst.Fields(16)
II3.SetFocus
II3.Text = rst.Fields(17)
MP3.SetFocus
MP3.Text = rst.Fields(18)
MS3.SetFocus
MS3.Text = rst.Fields(19)
RA3.SetFocus
RA3.Text = rst.Fields(20)
RM3.SetFocus
RM3.Text = rst.Fields(21)
End If

Case 4
If gvTreat(4) = False Then
TD4.SetFocus
TD4.Text = rst.Fields(3)
MSD4.SetFocus
MSD4.Text = rst.Fields(4)
PMTD4.SetFocus
PMTD4.Text = rst.Fields(5)
B4.SetFocus
B4.Text = rst.Fields(6)
HD4.SetFocus
HD4.Text = rst.Fields(7)
SHAD4.SetFocus
SHAD4.Text = rst.Fields(8)
HAC4.SetFocus
HAC4.Text = rst.Fields(9)
MAS4.SetFocus
MAS4.Text = rst.Fields(10)
DEH4.SetFocus
DEH4.Text = rst.Fields(11)
DH4.SetFocus
DH4.Text = rst.Fields(12)
LSH4.SetFocus
LSH4.Text = rst.Fields(13)
SA4.SetFocus
SA4.Text = rst.Fields(14)
IP4.SetFocus
IP4.Text = rst.Fields(15)
IS4.SetFocus
IS4.Text = rst.Fields(16)
II4.SetFocus
II4.Text = rst.Fields(17)
MP4.SetFocus
MP4.Text = rst.Fields(18)
MS4.SetFocus
MS4.Text = rst.Fields(19)
RA4.SetFocus
RA4.Text = rst.Fields(20)
RM4.SetFocus
RM4.Text = rst.Fields(21)

End If

Case 5
If gvTreat(5) = False Then
    TD5.SetFocus
    TD5.Text = rst.Fields(3)
    MSD5.SetFocus
    MSD5.Text = rst.Fields(4)
    PMTD5.SetFocus
    PMTD5.Text = rst.Fields(5)
B5.SetFocus
B5.Text = rst.Fields(6)
HD5.SetFocus
HD5.Text = rst.Fields(7)
SHAD5.SetFocus
SHAD5.Text = rst.Fields(8)
HAC5.SetFocus
HAC5.Text = rst.Fields(9)
MAS5.SetFocus
MAS5.Text = rst.Fields(10)
DEH5.SetFocus
DEH5.Text = rst.Fields(11)
DH5.SetFocus
DH5.Text = rst.Fields(12)
LSH5.SetFocus
LSH5.Text = rst.Fields(13)
SA5.SetFocus
SA5.Text = rst.Fields(14)
IP5.SetFocus
IP5.Text = rst.Fields(15)
IS5.SetFocus
IS5.Text = rst.Fields(16)
II5.SetFocus
II5.Text = rst.Fields(17)
MP5.SetFocus
MP5.Text = rst.Fields(18)
MS5.SetFocus
MS5.Text = rst.Fields(19)
RA5.SetFocus
RA5.Text = rst.Fields(20)
RM5.SetFocus
RM5.Text = rst.Fields(21)
End If
Case 6
If gvTreat(6) = False Then
    TD6.SetFocus
    TD6.Text = rst.Fields(3)
    MSD6.SetFocus
    MSD6.Text = rst.Fields(4)
    PMTD6.SetFocus
    PMTD6.Text = rst.Fields(5)
    B6.SetFocus
    B6.Text = rst.Fields(6)
    HD6.SetFocus
    HD6.Text = rst.Fields(7)
    SHAD6.SetFocus
    SHAD6.Text = rst.Fields(8)
    HAC6.SetFocus
    HAC6.Text = rst.Fields(9)
    MAS6.SetFocus
    MAS6.Text = rst.Fields(10)
    DEH6.SetFocus
    DEH6.Text = rst.Fields(11)
    DH6.SetFocus
    DH6.Text = rst.Fields(12)
    LSH6.SetFocus
    LSH6.Text = rst.Fields(13)
    SA6.SetFocus
End If

Case 7
If gvTreat(7) = False Then
    TD7.SetFocus
    TD7.Text = rst.Fields(3)
    MSD7.SetFocus
    MSD7.Text = rst.Fields(4)
    PMTD7.SetFocus
    PMTD7.Text = rst.Fields(5)
    B7.SetFocus
    B7.Text = rst.Fields(6)
    HD7.SetFocus
    HD7.Text = rst.Fields(7)
    SHAD7.SetFocus
SHAD7.Text = rst.Fields(8)
HAC7.SetFocus
HAC7.Text = rst.Fields(9)
MAS7.SetFocus
MAS7.Text = rst.Fields(10)
DEH7.SetFocus
DEH7.Text = rst.Fields(11)
DH7.SetFocus
DH7.Text = rst.Fields(12)
LSH7.SetFocus
LSH7.Text = rst.Fields(13)
SA7.SetFocus
SA7.Text = rst.Fields(14)
IP7.SetFocus
IP7.Text = rst.Fields(15)
IS7.SetFocus
IS7.Text = rst.Fields(16)
II7.SetFocus
II7.Text = rst.Fields(17)
MP7.SetFocus
MP7.Text = rst.Fields(18)
MS7.SetFocus
MS7.Text = rst.Fields(19)
RA7.SetFocus
RA7.Text = rst.Fields(20)
RM7.SetFocus
RM7.Text = rst.Fields(21)
End If
Case Else
MsgBox "Error in recordset for treatment days."
End Select
rst.MoveNext
Loop
cnn.Close
End Sub

7. Summary Chart Display Form

Option Compare Database
Private Sub Form_Load()
    DoCmd.OpenForm "frmSummary"
End Sub

Private Sub ButtonClear_Click()
    ChkTD.Value = False
    ChkMSD.Value = False
    ChkPMTD.Value = False
    ChkB.Value = False
    ChkHD.Value = False
    ChkSHAD.Value = False
    ChkHC.Value = False
    ChkMAS.Value = False
    ChkDEH.Value = False
    ChkDH.Value = False
    ChkLSH.Value = False
    ChkSA.Value = False
    ChkIP.Value = False
    ChkIS.Value = False
ChkII.Value = False
ChkMP.Value = False
ChkMM.Value = False
ChkRA.Value = False
ChkRM.Value = False
If gvMode = False Then
    Title.Caption = "Headache Treatment Progress for Individual"
    WhatNum.Caption = "Patient Num:"
    Number.SetFocus
    Number.Text = gvPatientNum
Else
    Title.Caption = "Headache Treatment Progress for Group"
    WhatNum.Caption = "Group Num:"
    Number.SetFocus
    Number.Text = gvGroupNum
End If
MSGraph0.RowSource = "Select * From CORE30 Where CORE30.Patient = 9999;"
End Sub

Private Sub ButtonDisplay_Click()
    Dim strSQL As String

    If gvMode = False Then
        strSQL = "SELECT RESULT.PHASE "
        If ChkTD.Value = True Then
            strSQL = strSQL & ", RESULT.TREAT_DAYS "
        End If
        If ChkMSD.Value = True Then

        End If
    End If
strSQL = strSQL & ", RESULT.MISS_DAYS "
End If
If ChkPMTD.Value = True Then
    strSQL = strSQL & ", RESULT.PMT_DAYS "
End If
If ChkB.Value = True Then
    strSQL = strSQL & ", RESULT.BONUS "
End If
If ChkB.Value = True Then
    strSQL = strSQL & ", RESULT.HADAYS "
End If
If ChkB.Value = True Then
    strSQL = strSQL & ", Result.SHADAYS "
End If
If ChkB.Value = True Then
    strSQL = strSQL & ", RESULT.HACOUNT "
End If
If ChkB.Value = True Then
    strSQL = strSQL & ", RESULT.MAX_AVGSEV "
End If
If ChkB.Value = True Then
    strSQL = strSQL & ", RESULT.DEHOURS "
End If
If ChkB.Value = True Then
    strSQL = strSQL & ", RESULT.DHOURS "
End If
If ChkB.Value = True Then
    strSQL = strSQL & ", RESULT.LSHOURS "
End If
If ChkSA.Value = True Then
    strSQL = strSQL & " , RESULT.ANALGESIC"
End If
If ChkIP.Value = True Then
    strSQL = strSQL & " , RESULT.IMPILL"
End If
If ChkIS.Value = True Then
    strSQL = strSQL & " , RESULT.IMSPRAY"
End If
If ChkII.Value = True Then
    strSQL = strSQL & " , RESULT.IMINJ"
End If
If ChkMP.Value = True Then
    strSQL = strSQL & " , RESULT.MAXTPILL"
End If
If ChkMM.Value = True Then
    strSQL = strSQL & " , RESULT.MAXTMELT"
End If
If ChkRA.Value = True Then
    strSQL = strSQL & " , RESULT.REGLAN"
End If
If ChkRM.Value = True Then
    strSQL = strSQL & " , RESULT.RESCUE"
End If

strSQL = strSQL & " FROM CORE30 As RESULT " & _
    "WHERE RESULT.PATIENT = " & gvPatientNum & " And 
RESULT.HATYPE = " & gvHeadache
If gvTreat(0) = True Then
    strSQL = strSQL & " And RESULT.PHASE <> 0 "
End If
If gvTreat(1) = True Then
    strSQL = strSQL & " And RESULT.PHASE <> 1 "
End If
If gvTreat(2) = True Then
    strSQL = strSQL & " And RESULT.PHASE <> 2 "
End If
If gvTreat(3) = True Then
    strSQL = strSQL & " And RESULT.PHASE <> 3 "
End If
If gvTreat(4) = True Then
    strSQL = strSQL & " And RESULT.PHASE <> 4 "
End If
If gvTreat(5) = True Then
    strSQL = strSQL & " And RESULT.PHASE <> 5 "
End If
If gvTreat(6) = True Then
    strSQL = strSQL & " And RESULT.PHASE <> 6 "
End If
If gvTreat(7) = True Then
    strSQL = strSQL & " And RESULT.PHASE <> 7 "
End If
strSQL = strSQL & " ORDER BY RESULT.PHASE;"
Else
    strSQL = "SELECT Distinct Result.PHASE "
If ChkTD.Value = True Then
    strSQL = strSQL & ", Avg(Result.TREAT_DAYS) As TREAT_DAYS "
If ChkMSD.Value = True Then
    strSQL = strSQL & ", Avg(Result.MISS_DAYS) As MISS_DAYS"
End If
If ChkPMTD.Value = True Then
    strSQL = strSQL & ", Avg(Result.PMT_DAYS) As PMTDAYS"
End If
If ChkB.Value = True Then
    strSQL = strSQL & ", Avg(Result.BONUS) As BONUS"
End If
If ChkHD.Value = True Then
    strSQL = strSQL & ", Avg(Result.HADAYS) As HADAYS"
End If
If ChkSHAD.Value = True Then
    strSQL = strSQL & ", Avg(Result.SHADAYS) As SHADAYS"
End If
If ChkHC.Value = True Then
    strSQL = strSQL & ", Avg(Result.HACOUNT) As HACOUNT"
End If
If ChkMAS.Value = True Then
    strSQL = strSQL & ", Avg(Result.MAX_AVGSEV) As MAX_AVGSEV"
End If
If ChkDEH.Value = True Then
    strSQL = strSQL & ", Avg(Result.DEHOURS) As DEHOURS"
End If
If ChkDH.Value = True Then
    strSQL = strSQL & ", Avg(Result.DHOURS) As DHOURS"
End If
If ChkLSH.Value = True Then
strSQL = strSQL & ", Avg(Result.LSHOURS) As LSHOURS"
End If
If ChkSA.Value = True Then
    strSQL = strSQL & ", Avg(Result.ANALGESIC) As ANALGESIC"
End If
If ChkIP.Value = True Then
    strSQL = strSQL & ", Avg(Result.IMPILL) As IMPILL"
End If
If ChkIS.Value = True Then
    strSQL = strSQL & ", Avg(Result.IMSPRAY) As IMSPRAY"
End If
If ChkII.Value = True Then
    strSQL = strSQL & ", Avg(Result.IMINJ) As IMINJ"
End If
If ChkMP.Value = True Then
    strSQL = strSQL & ", Avg(Result.MAXTPILL) As MAXTPILL"
End If
If ChkMM.Value = True Then
    strSQL = strSQL & ", Avg(Result.MAXTMELT) As MAXTMELT"
End If
If ChkRA.Value = True Then
    strSQL = strSQL & ", Avg(Result.REGLAN) As REGLAN"
End If
If ChkRM.Value = True Then
    strSQL = strSQL & ", Avg(Result.RESCUE) As RESCUE"
End If

strSQL = strSQL & " FROM CORE30 AS Result, SECRET_TBL As SecretTbl"
& WHERE SecretTbl.PATIENT = Result.PATIENT And
SecretTbl.GROUP
= " & gvGroupNum & " And Result.HATYPE = " & gvHeadache

If gvTreat(0) = True Then
    strSQL = strSQL & " And Result.PHASE <> 0 "
End If
If gvTreat(1) = True Then
    strSQL = strSQL & " And Result.PHASE <> 1 "
End If
If gvTreat(2) = True Then
    strSQL = strSQL & " And Result.PHASE <> 2 "
End If
If gvTreat(3) = True Then
    strSQL = strSQL & " And Result.PHASE <> 3 "
End If
If gvTreat(4) = True Then
    strSQL = strSQL & " And Result.PHASE <> 4 "
End If
If gvTreat(5) = True Then
    strSQL = strSQL & " And Result.PHASE <> 5 "
End If
If gvTreat(6) = True Then
    strSQL = strSQL & " And Result.PHASE <> 6 "
End If
If gvTreat(7) = True Then
    strSQL = strSQL & " And Result.PHASE <> 7 "
End If
strSQL = strSQL & " Group By Result.PHASE ORDER BY Result.PHASE;"
End If

MSGraph0.RowSource = strSQL
End Sub

Private Sub Form_Load()
    ChkTD.Value = False
    ChkMSD.Value = False
    ChkPMTD.Value = False
    ChkB.Value = False
    ChkHD.Value = False
    ChkSHAD.Value = False
    ChkHC.Value = False
    ChkMAS.Value = False
    ChkDEH.Value = False
    ChkDH.Value = False
    ChkLSH.Value = False
    ChkSA.Value = False
    ChkIP.Value = False
    ChkIS.Value = False
    ChkII.Value = False
    ChkMP.Value = False
    ChkMM.Value = False
    ChkRA.Value = False
    ChkRM.Value = False

    HAType.SetFocus
    If gvHeadache = 1 Then
HAType.Text = "Migraine"
ElseIf gvHeadache = 5 Then
    HAType.Text = "All Headache"
ElseIf gvHeadache = 2 Then
    HAType.Text = "Tension"
ElseIf gvHeadache = 3 Then
    HAType.Text = "Other Headache"
ElseIf gvHeadache = 4 Then
    HAType.Text = "Unknown Headache"
Else
    MsgBox ("Error in headache type selection!")
    Exit Sub
End If

If gvMode = False Then
    Title.Caption = "Headache Treatment Progress for Individual"
    WhatNum.Caption = "Patient Num:"
    Number.SetFocus
    Number.Text = gvPatientNum
Else
    Title.Caption = "Headache Treatment Progress for Group"
    WhatNum.Caption = "Group Num:"
    Number.SetFocus
    Number.Text = gvGroupNum
End If

MSGraph0.RowSource = "Select * from CORE30
    Where CORE30.PATIENT = 9999 ;"
End Sub

Private Sub ButtonClose_Click()
    On Error GoTo Err_ButtonClose_Click
    DoCmd.Close
    Exit_ButtonClose_Click:
    Exit Sub

Err_ButtonClose_Click:
    MsgBox Err.Description
    Resume Exit_ButtonClose_Click
End Sub

7.3 SQL Queries Used in Diary Database (in alphabetical order)

AppendContact
INSERT INTO CONTACT_MASTER
SELECT CONTACT_COPY.*
FROM CONTACT_COPY;

AppendCore1
INSERT INTO CORE30 (PATIENT, PHASE, HATYPE, TREAT_DAYS,
MISS_DAYS, PMT_DAYS, BONUS, HADAYS, SHADAYS, HACCOUNT,
MAX_AVGSEV, DEHOURS, DHOURS, LSHOURS, ANALGESIC, IMPILL,
IMSPRAY, IMINJ, MAXTPILL, MAXTMELT, REGLAN, RESCUE )
SELECT ALLHA.PATIENT, ALLHA.PHASE, 1 AS Expr1,
ALLHA.TREAT_DAYS,
[ALLHA].[MISS_DAYS]/[ALLHA].[TREAT_DAYS]*30 AS Expr2,
[ALLHA].[PMT_DAYS]/[ALLHA].[TREAT_DAYS]*30 AS Expr3, 0 AS Expr4,
[MIGRAINE].[HADAYS]/[ALLHA].[TREAT_DAYS]*30 AS Expr5,
[MIGRAINE].[SHADAYS]/[ALLHA].[TREAT_DAYS]*30 AS Expr18,
[MIGRAINE].[HACOUNT]/[ALLHA].[TREAT_DAYS]*30 AS Expr6,
MIGRAINE.MAX_AVGSEV,
[MIGRAINE].[DEHOURS]/[ALLHA].[TREAT_DAYS]*30 AS Expr7,
[MIGRAINE].[DHOURS]/[ALLHA].[TREAT_DAYS]*30 AS Expr8,
[MIGRAINE].[LSHOURS]/[ALLHA].[TREAT_DAYS]*30 AS Expr9,
[MIGRAINE].[ANALGESIC]/[ALLHA].[TREAT_DAYS]*30 AS Expr10,
[MIGRAINE].[IMPILL]/[ALLHA].[TREAT_DAYS]*30 AS Expr11,
[MIGRAINE].[IMSPRAY]/[ALLHA].[TREAT_DAYS]*30 AS Expr12,
[MIGRAINE].[IMINJ]/[ALLHA].[TREAT_DAYS]*30 AS Expr13,
[MIGRAINE].[MAXTPILL]/[ALLHA].[TREAT_DAYS]*30 AS Expr14,
[MIGRAINE].[MAXTMELT]/[ALLHA].[TREAT_DAYS]*30 AS Expr15,
[MIGRAINE].[REGLAN]/[ALLHA].[TREAT_DAYS]*30 AS Expr16,
[MIGRAINE].[RESCUE]/[ALLHA].[TREAT_DAYS]*30 AS Expr17
FROM ALLHA, MIGRAINE, CONTACT_COPY
WHERE ALLHA.PATIENT = CONTACT_COPY.PATIENT AND
MIGRAINE.PATIENT = CONTACT_COPY.PATIENT AND ALLHA.PHASE =
CONTACT_COPY.PHASE AND
MIGRAINE.PHASE = CONTACT_COPY.PHASE;

AppendCore2
INSERT INTO CORE30 (PATIENT, PHASE, HATYPE, TREAT_DAYS,
MISS_DAYS, PMT_DAYS, HADAYS, SHADAYS, HACOUNT, BONUS,
MAX_AVGSEV, DEHOURS, DHOURS, LSHOURS, ANALGESIC, IMPILL,
IMSPRAY, IMINJ, MAXTPILL, MAXTMELT, REGLAN, RESCUE )
SELECT ALLHA.PATIENT, ALLHA.PHASE, 2 AS Expr1,
ALLHA.TREAT_DAYS,
[ALLHA].[MISS_DAYS]/[ALLHA].[TREAT_DAYS]*30 AS Expr2,
[ALLHA].[PMT_DAYS]/[ALLHA].[TREAT_DAYS]*30 AS Expr3,
[TENSION].[HADAYS]/[ALLHA].[TREAT_DAYS]*30 AS Expr4,
[TENSION].[SHADAYS]/[ALLHA].[TREAT_DAYS]*30 AS Expr18,
[TENSION].[HACOUNT]/[ALLHA].[TREAT_DAYS]*30 AS Expr5, 0 AS Expr6,
TENSION.MAX_AVGSEV,
[TENSION].[DEHOURS]/[ALLHA].[TREAT_DAYS]*30 AS Expr7,
[TENSION].[DHOURS]/[ALLHA].[TREAT_DAYS]*30 AS Expr8,
[TENSION].[LSHOURS]/[ALLHA].[TREAT_DAYS]*30 AS Expr9,
[TENSION].[ANALGESIC]/[ALLHA].[TREAT_DAYS]*30 AS Expr10,
[TENSION].[IMPILL]/[ALLHA].[TREAT_DAYS]*30 AS Expr11,
[TENSION].[IMSPRAY]/[ALLHA].[TREAT_DAYS]*30 AS Expr12,
[TENSION].[IMINJ]/[ALLHA].[TREAT_DAYS]*30 AS Expr13,
[TENSION].[MAXTPILL]/[ALLHA].[TREAT_DAYS]*30 AS Expr14,
[TENSION].[MAXTMELT]/[ALLHA].[TREAT_DAYS]*30 AS Expr15,
[TENSION].[REGLAN]/[ALLHA].[TREAT_DAYS]*30 AS Expr16,
[TENSION].[RESCUE]/[ALLHA].[TREAT_DAYS]*30 AS Expr17
FROM ALLHA, TENSION, CONTACT COPY
WHERE ALLHA.PATIENT = CONTACT_COPY.PATIENT AND
TENSION.PATIENT = CONTACT_COPY.PATIENT AND ALLHA.PHASE =
CONTACT_COPY.PHASE AND
TENSION.PHASE = CONTACT_COPY.PHASE;
Insert into CORE30 (PATIENT, PHASE, HATYPE, TREAT_DAYS, MISS_DAYS, PMT_DAYS, HADAYS, SHADAYS, HACOUNT, BONUS, MAX_AVGSEV, DEHOURS, D_HOURS, LSHOURS, ANALGESIC, IMPILL, IMSpray, IMINJ, MAXTPILL, MAXTMELT, REGLAN, RESCUE) select ALLHA.PATIENT, ALLHA.PHASE, 3 as Expr1, ALLHA.TREAT_DAYS, [ALLHA].[MISS_DAYS]/[ALLHA].[TREAT_DAYS]*30 as Expr2, [ALLHA].[PMT_DAYS]/[ALLHA].[TREAT_DAYS]*30 as Expr3, [OTHERHA].[HADAYS]/[ALLHA].[TREAT_DAYS]*30 as Expr4, [OTHERHA].[SHADAYS]/[ALLHA].[TREAT_DAYS]*30 as Expr18, [OTHERHA].[HACOUNT]/[ALLHA].[TREAT_DAYS]*30 as Expr5, 0 as Expr6, OTHERHA.MAX_AVGSEV, [OTHERHA].[DEHOURS]/[ALLHA].[TREAT_DAYS]*30 as Expr7, [OTHERHA].[D_HOURS]/[ALLHA].[TREAT_DAYS]*30 as Expr8, [OTHERHA].[LSHOURS]/[ALLHA].[TREAT_DAYS]*30 as Expr9, [OTHERHA].[ANALGESIC]/[ALLHA].[TREAT_DAYS]*30 as Expr10, [OTHERHA].[IMPELL]/[ALLHA].[TREAT_DAYS]*30 as Expr11, [OTHERHA].[IMSPRAY]/[ALLHA].[TREAT_DAYS]*30 as Expr12, [OTHERHA].[IMINJ]/[ALLHA].[TREAT_DAYS]*30 as Expr13, [OTHERHA].[MAXTPILL]/[ALLHA].[TREAT_DAYS]*30 as Expr14, [OTHERHA].[MAXTMELT]/[ALLHA].[TREAT_DAYS]*30 as Expr15, [OTHERHA].[REGLAN]/[ALLHA].[TREAT_DAYS]*30 as Expr16, [OTHERHA].[RESCUE]/[ALLHA].[TREAT_DAYS]*30 as Expr17 from ALLHA, OTHERHA, CONTACT_COPY where ALLHA.PATIENT = CONTACT_COPY.PATIENT and OTHERHA.PATIENT = CONTACT_COPY.PATIENT and ALLHA.PHASE = CONTACT_COPY.PHASE and
OTHERHA.PHASE = CONTACT_COPY.PHASE;

**AppendCore4**

```sql
INSERT INTO CORE30 (PATIENT, PHASE, HATYPE, TREAT_DAYS, MISS_DAYS, PMT_DAYS, HADAYS, SHADAYS, HACOUNT, BONUS, MAX_AVGSEV, DEHOURS, DHOURS, LSHOURS, ANALGESIC, IMPELL, IMSPRAY, IMINJ, MAXTPILL, MAXTMELT, REGLAN, RESCUE )
SELECT ALLHA.PATIENT, ALLHA.PHASE, 1 AS Expr1,
ALLHA.TREAT_DAYS,
ALLHA.[MISS_DAYS]/ALLHA.[TREAT_DAYS]*30 AS Expr2,
ALLHA.[PMT_DAYS]/ALLHA.[TREAT_DAYS]*30 AS Expr3,
UNKNOWN.[HADAYS]/ALLHA.[TREAT_DAYS]*30 AS Expr4,
UNKNOWN.[SHADAYS]/ALLHA.[TREAT_DAYS]*30 AS Expr8,
UNKNOWN.[HACOUNT]/ALLHA.[TREAT_DAYS]*30 AS Expr5, 0 AS Expr6, UNKNOWN.MAX_AVGSEV,
UNKNOWN.[DEHOURS]/ALLHA.[TREAT_DAYS]*30 AS Expr7,
UNKNOWN.[DHOURS]/ALLHA.[TREAT_DAYS]*30 AS Expr8,
UNKNOWN.[LSHOURS]/ALLHA.[TREAT_DAYS]*30 AS Expr9,
UNKNOWN.[ANALGESIC]/ALLHA.[TREAT_DAYS]*30 AS Expr10,
UNKNOWN.[IMPELL]/ALLHA.[TREAT_DAYS]*30 AS Expr11,
UNKNOWN.[IMSPRAY]/ALLHA.[TREAT_DAYS]*30 AS Expr12,
UNKNOWN.[IMINJ]/ALLHA.[TREAT_DAYS]*30 AS Expr13,
UNKNOWN.[MAXTPILL]/ALLHA.[TREAT_DAYS]*30 AS Expr14,
UNKNOWN.[MAXTMELT]/ALLHA.[TREAT_DAYS]*30 AS Expr15,
UNKNOWN.[REGLAN]/ALLHA.[TREAT_DAYS]*30 AS Expr16,
UNKNOWN.[RESCUE]/ALLHA.[TREAT_DAYS]*30 AS Expr17
FROM ALLHA, UNKNOWN, CONTACT_COPY
```
WHERE ALLHA.PATIENT = CONTACT_COPY.PATIENT AND
UNKNOWN.PATIENT = CONTACT_COPY.PATIENT AND ALLHA.PHASE =
CONTACT.Copy.PHASE AND
UNKNOWN.PHASE = CONTACT_COPY.PHASE;

.AppendCore5
INSERT INTO CORE30 (PATIENT, PHASE, HATYPE, TREAT_DAYS,
MISS_DAYS, PMT_DAYS, BONUS, HADAYS, SHADAYS, HACCOUNT,
MAX_AVGSEV, DEHOURS, DHOURS, LSHOURS, ANALGESIC, IMPILL,
IMSPRAY, IMINJ, MAXTPILL, MAXTMELT, REGLAN, RESCUE )
SELECT ALLHA.PATIENT, ALLHA.PHASE, 5 AS Expr1,
ALLHA.TREAT_DAYS,
ALLHA.[MISS_DAYS]/ALLHA.[TREAT_DAYS]*30 AS Expr2,
ALLHA.[PMT_DAYS]/ALLHA.[TREAT_DAYS]*30 AS Expr3,
ALLHA.BONUS, ALLHA.[HADAYS]/ALLHA.[TREAT_DAYS]*30 AS Expr5,
ALLHA.[SHADAYS]/ALLHA.[TREAT_DAYS]*30 AS Expr18,
ALLHA.[HACCOUNT]/ALLHA.[TREAT_DAYS]*30 AS Expr6,
ALLHA.MAX_AVGSEV, ALLHA.[DEHOURS]/ALLHA.[TREAT_DAYS]*30 AS Expr7,
ALLHA.[DHOURS]/ALLHA.[TREAT_DAYS]*30 AS Expr8,
ALLHA.[LSHOURS]/ALLHA.[TREAT_DAYS]*30 AS Expr9,
ALLHA.[ANALGESIC]/ALLHA.[TREAT_DAYS]*30 AS Expr10,
ALLHA.[IMPELL]/ALLHA.[TREAT_DAYS]*30 AS Expr11,
ALLHA.[IMSPRAY]/ALLHA.[TREAT_DAYS]*30 AS Expr12,
ALLHA.[IMINJ]/ALLHA.[TREAT_DAYS]*30 AS Expr13,
ALLHA.[MAXTPILL]/ALLHA.[TREAT_DAYS]*30 AS Expr14,
ALLHA.[MAXTMELT]/ALLHA.[TREAT_DAYS]*30 AS Expr15,
ALLHA.[REGLAN]/ALLHA.[TREAT_DAYS]*30 AS Expr16,
ALLHA.[RESCUE]/ALLHA.[TREAT_DAYS]*30 AS Expr17
FROM ALLHA, CONTACT_COPY
WHERE ALLHA.PATIENT = CONTACT_COPY.PATIENT AND
ALLHA.PHASE = CONTACT_COPY.PHASE;

**AppendMainData**

INSERT INTO MAINDATA_MASTER
SELECT MAINDATA_COPY.*
FROM MAINDATA_COPY;

**AppendMigraine**

INSERT INTO MIGRAINE (PATIENT, PHASE, MAX_AVGSEV, HADAYS, SHADAYS, HACOUNT, DEHOURS, DHOURS, LSHOURS, ANALGESIC, IMPILL, IMSPRAY, IMINJ, MAXTPILL, MAXTMELT, REGLAN, RESCUE )
SELECT DISTINCT MAINDATA_COPY.PATIENT, 
MAINDATA_COPY.PHASE, Avg([MAINDATA_COPY][MAX_SEV]) AS Expr1,
Sum([MAINDATA_COPY][DAYS]) AS Expr2,
Sum([MAINDATA_COPY][SDAYS]) AS Expr3,
Count([MAINDATA_COPY][REF]) AS Expr4,
Sum([MAINDATA_COPY][DEHOURS]) AS Expr5,
Sum([MAINDATA_COPY][DHOURS]) AS Expr6,
Sum([MAINDATA_COPY][MISS_D]) AS Expr7,
Sum([MAINDATA_COPY][ANALGESIC]) AS Expr8,
Sum([MAINDATA_COPY][IMIT_PILL]) AS Expr9,
Sum([MAINDATA_COPY][IMIT_SPRAY]) AS Expr10,
Sum([MAINDATA_COPY][IMIT_INJ]) AS Expr11,
Sum([MAINDATA_COPY][MAXTPILL]) AS Expr12,
Sum([MAINDATA_COPY][MAXTMELT]) AS Expr13,
Sum([MAINDATA_COPY][REGLAN]) AS Expr14
FROM MAINDATA_COPY
WHERE MAINDATA_COPY.HD_TYPE=1
GROUP BY MAINDATA_COPY.PATIENT, MAINDATA_COPY.PHASE;

AppendOtherHA

INSERT INTO OTHERHA (PATIENT, PHASE, MAX_AVGSEV, HADAYS,
SHADAYS, HACCOUNT, DEHOURS, DHOURS, LSHOURS, ANALGESIC,
IMPILL, IMSPRAY, IMINJ, MAXTPILL, MAXTMELT, REGLAN, RESCUE )
SELECT DISTINCT MAINDATA_COPY.PATIENT,
MAINDATA_COPY.PHASE, Avg([MAINDATA_COPY]![MAX_SEV]) AS Expr1,
Sum([MAINDATA_COPY]![DAYS]) AS Expr2,
Sum([MAINDATA_COPY]![SDAYS]) AS Expr15,
Count([MAINDATA_COPY]![REF]) AS Expr3,
Sum([MAINDATA_COPY]![DEHOURS]) AS Expr4,
Sum([MAINDATA_COPY]![DHOURS]) AS Expr5,
Sum([MAINDATA_COPY]![MISS_D]) AS Expr6,
Sum([MAINDATA_COPY]![ANALGESIC]) AS Expr7,
Sum([MAINDATA_COPY]![IMIT_PILL]) AS Expr8,
Sum([MAINDATA_COPY]![IMIT_SPRAY]) AS Expr9,
Sum([MAINDATA_COPY]![IMIT_INJ]) AS Expr10,
Sum([MAINDATA_COPY]![MAXT_PILL]) AS Expr11,
Sum([MAINDATA_COPY]![MAXT_MELT]) AS Expr12,
Sum([MAINDATA_COPY]![REGLAN]) AS Expr13,
Sum([MAINDATA_COPY]![RESCUE]) AS Expr14
FROM MAINDATA_COPY
WHERE MAINDATA_COPY.HD_TYPE =3
GROUP BY MAINDATA_COPY.PATIENT, MAINDATA_COPY.PHASE;
AppendTension

INSERT INTO TENSION (PATIENT, PHASE, MAX_AVGSEV, HADAYS, SHADAYS, HACOUNT, DEHOURS, DHOURS, LSHOURS, ANALGESIC, IMPILL, IMSPRAY, IMINJ, MAXTPILL, MAXTMELT, REGLAN, RESCUE) 
SELECT MAINDATA_COPY.PATIENT, MAINDATA_COPY.PHASE, 
Avg([MAINDATA_COPY][MAX_SEV]) AS Expr1, 
Sum([MAINDATA_COPY][DAYS]) AS Expr2, 
Sum([MAINDATA_COPY][SDAYS]) AS Expr15, 
Count([MAINDATA_COPY][REF]) AS Expr3, 
Sum([MAINDATA_COPY][DEHOURS]) AS Expr4, 
Sum([MAINDATA_COPY][DHOURS]) AS Expr5, 
Sum([MAINDATA_COPY][MISS_D]) AS Expr6, 
Sum([MAINDATA_COPY][ANALGESIC]) AS Expr7, 
Sum([MAINDATA_COPY][IMIT_PILL]) AS Expr8, 
Sum([MAINDATA_COPY][IMIT_SPRAY]) AS Expr9, 
Sum([MAINDATA_COPY][IMIT_INJ]) AS Expr10, 
Sum([MAINDATA_COPY][MAXT_PILL]) AS Expr11, 
Sum([MAINDATA_COPY][MAXT_MELT]) AS Expr12, 
Sum([MAINDATA_COPY][REGLAN]) AS Expr13, 
Sum([MAINDATA_COPY][RESCUE]) AS Expr14 
FROM MAINDATA_COPY 
WHERE MAINDATA_COPY.HD_TYPE=2 
GROUP BY MAINDATA_COPY.PATIENT, MAINDATA_COPY.PHASE;

AppendUnknown

INSERT INTO [UNKNOWN] (PATIENT, PHASE, MAX_AVGSEV, HADAYS, SHADAYS, HACOUNT, DEHOURS, DHOURS, LSHOURS, ANALGESIC, IMPILL, IMSPRAY, IMINJ, MAXTPILL, MAXTMELT, REGLAN, RESCUE)
SELECT DISTINCT MAINDATA_COPY.PATIENT,
MAINDATA_COPY.PHASE, Avg([MAINDATA_COPY]![MAX_SEV]) AS Expr1,
Sum([MAINDATA_COPY]![DAYS]) AS Expr2,
Sum([MAINDATA_COPY]![SDAYS]) AS Expr15,
Count([MAINDATA_COPY]![REF]) AS Expr3,
Sum([MAINDATA_COPY]![DEHOURS]) AS Expr4,
Sum([MAINDATA_COPY]![DHOURLS]) AS Expr5,
Sum([MAINDATA_COPY]![MISS_D]) AS Expr6,
Sum([MAINDATA_COPY]![ANALGESIC]) AS Expr7,
Sum([MAINDATA_COPY]![IMIT_PILL]) AS Expr8,
Sum([MAINDATA_COPY]![IMIT_SPRAY]) AS Expr9,
Sum([MAINDATA_COPY]![IMIT_INJ]) AS Expr10,
Sum([MAINDATA_COPY]![MAXT_PILL]) AS Expr11,
Sum([MAINDATA_COPY]![MAXT_MELT]) AS Expr12,
Sum([MAINDATA_COPY]![REGLAN]) AS Expr13,
Sum([MAINDATA_COPY]![RESCUE]) AS Expr14
FROM MAINDATA_COPY
WHERE MAINDATA_COPY.HD_TYPE=4
GROUP BY MAINDATA_COPY.PATIENT, MAINDATA_COPY.PHASE;
CopyContact

INSERT INTO CONTACT_COPY (REF, PATIENT, PHASE, [DATE], GENDER, MENSTR, DAILYMED, CONTIMES, HADAY, BONUS) 
SELECT CONTACT.REF, CONTACT.PATIENT, CONTACT.PHASE, 
CONTACT.DATE, CONTACT.GENDER, IIf(IsNull([CONTACT]![MENSTR]),-1,[CONTACT]![MENSTR]) AS Expr1, IIf(IsNull([CONTACT]![DAILYMED]),-1,[CONTACT]![DAILYMED]) AS Expr2, CONTACT.CONTIMES, 
CONTACT.HADAY, CONTACT.BONUS 
FROM CONTACT;

CopyMainData

INSERT INTO MAINDATA_COPY (REF, PATIENT, PHASE, INPUT_DATE, INPUT_TIME, START_DATE, START_TIME, END_DATE, END_TIME, HD_TYPE, MAX_SEV, NAUSEA, PHONO, PHOTO, VOMITING, ANALGESIC, IMIT_PILL, IMIT_SPRAY, IMIT_INJ, MAXT_PILL, MAXT_MELT, REGLAN, RESCUE, MISS_A, DONE_A, MISS_B, DONE_B, MISS_C, DONE_C, MISS_D) 
SELECT MAINDATA.REF, MAINDATA.PATIENT, MAINDATA.PHASE, 
MAINDATA.INPUT_DATE, MAINDATA.INPUT_TIME, 
IIf(IsNull([MAINDATA]![START_DATE]),-1,[MAINDATA]![START_DATE]) AS Expr1, IIf(IsNull([MAINDATA]![START_TIME]),-1,[MAINDATA]![START_TIME]) AS Expr2, 
IIf(IsNull([MAINDATA]![END_DATE]),-1,[MAINDATA]![END_DATE]) AS Expr3, IIf(IsNull([MAINDATA]![END_TIME]),-1,[MAINDATA]![END_TIME]) AS Expr4, 
IIf(IsNull([MAINDATA]![HD_TYPE]),-1,[MAINDATA]![HD_TYPE]) AS Expr5, IIf(IsNull([MAINDATA]![MAX_SEV]),-1,[MAINDATA]![MAX_SEV]) AS Expr6, 
IIf(IsNull([MAINDATA]![NAUSEA]),-1,[MAINDATA]![NAUSEA]) AS Expr7, IIf(IsNull([MAINDATA]![PHONO]),-1,[MAINDATA]![PHONO]) AS Expr8, 
IIf(IsNull([MAINDATA]![PHOTO]),-1,[MAINDATA]![PHOTO]) AS Expr9, IIf(IsNull([MAINDATA]![VOMITING]),-1,[MAINDATA]![VOMITING]) AS Expr10 
FROM MAINDATA;
1. (MAINDATA)![VOMITING]) AS Expr10,
   Iff(IsNull((MAINDATA)![ANALGESIC]),-1,(MAINDATA)![ANALGESIC]) AS Expr11,
   Iff(IsNull((MAINDATA)![IMIT_PILL]),-1,(MAINDATA)![IMIT_PILL]) AS Expr12,
   If(IsNull((MAINDATA)![IMIT_SPRAY]),-1,(MAINDATA)![IMIT_SPRAY]) AS Expr13,
   Iff(IsNull((MAINDATA)![IMIT_INJ]),-1,(MAINDATA)![IMIT_INJ]) AS Expr14,
   Iff(IsNull((MAINDATA)![MAXT_PILL]),-1,(MAINDATA)![MAXT_PILL]) AS Expr15,
   If(IsNull((MAINDATA)![MAXT_MELT]),-1,(MAINDATA)![MAXT_MELT]) AS Expr16,
   If(IsNull((MAINDATA)![REGLAN]),-1,(MAINDATA)![REGLAN]) AS Expr17,
   If(IsNull((MAINDATA)![RESCUE]),-1,(MAINDATA)![RESCUE]) AS Expr18,
   If(IsNull((MAINDATA)![MISS_A]),-1,(MAINDATA)![MISS_A]) AS Expr19,
   If(IsNull((MAINDATA)![DONE_A]),-1,(MAINDATA)![DONE_A]) AS Expr20,
   If(IsNull((MAINDATA)![MISS_B]),-1,(MAINDATA)![MISS_B]) AS Expr21,
   If(IsNull((MAINDATA)![DONE_B]),-1,(MAINDATA)![DONE_B]) AS Expr22,
   If(IsNull((MAINDATA)![MISS_C]),-1,(MAINDATA)![MISS_C]) AS Expr23,
   If(IsNull((MAINDATA)![DONE_C]),-1,(MAINDATA)![DONE_C]) AS Expr24,
   If(IsNull((MAINDATA)![MISS_D]),-1,(MAINDATA)![MISS_D]) AS Expr25
FROM MAINDATA;

CopyMainDataStep2
UPDATE MAINDATA_COPY SET MAINDATA_COPY.DEHOURS =
   (MAINDATA_COPY)![MISS_A]+(MAINDATA_COPY)![MISS_B]+(MAINDATA_COPY)![MISS_C]+(MAINDATA_COPY)![MISS_D]+(MAINDATA_COPY)![DONE_A]+(MAINDATA_COPY)![DONE_B]+(MAINDATA_COPY)![DONE_C])/2,
MAINDATA_COPY.DHOURS =
   (MAINDATA_COPY)![MISS_A]+(MAINDATA_COPY)![MISS_B]+(MAINDATA_COPY)![MISS_C],
MAINDATA_COPY.DAYS =
   (MAINDATA_COPY)![END_DATE]-[MAINDATA_COPY]![START_DATE]+1,
MAINDATA_COPY.SDAYS =
IIf((MAINDATA_COPY)!(IMIT_PILL)+[MAINDATA_COPY]![IMIT_SPRAY]+[MAINDATA_COPY]![IMIT_INJ]+[MAINDATA_COPY]![MAXT_PILL]+[MAINDATA_COPY]![MAXT_MELT]+[MAINDATA_COPY]![MISS_A]+[MAINDATA_COPY]![DONE_A]+[MAINDATA_COPY]![MISS_B]+[MAINDATA_COPY]![DONE_B]+[MAINDATA_COPY]![MISS_C]+[MAINDATA_COPY]![DONE_C]+[MAINDATA_COPY]![MISS_D]>0,[MAINDATA_COPY]![END_DATE]-[MAINDATA_COPY]![START_DATE]+1,0);

EmptyContact
DELETE CONTACT_COPY.*
FROM CONTACT_COPY;

EmptyMainData_Copy
DELETE MAINDATA_COPY.*
FROM MAINDATA_COPY;

ExportAllha
SELECT ALLHA.* INTO [dBASE 5.0;Database=C:\Diary\Files2Export].Wallha
FROM ALLHA;

ExportContact
SELECT CONTACT_MASTER.* INTO
[dBASE5.0;Database=C:\Diary\Files2Export].Wcontact
FROM CONTACT_MASTER;

ExportCore30
SELECT CORE30.* INTO [dBASE 5.0;Database=C:\Diary\Files2Export].Wcore30
FROM CORE30;
ExportMainData
SELECT MAINDATA_MASTER.* INTO [dBASE5.0;Database=C:\Diary\Files2Export].Wmaindat
FROM MAINDATA_MASTER;

ExportMigraine
SELECT MIGRAINE.* INTO [dBASE 5.0;Database=C:\Diary\Files2Export].Wmigrain
FROM MIGRAINE;

ExportOtherHA
SELECT OTHERHA.* INTO [dBASE 5.0;Database=C:\Diary\Files2Export].Wother
FROM OTHERHA;

ExportPatient
SELECT PATIENT_INFO.* INTO [dBASE 5.0;Database=C:\Diary\Files2Export].Wpatient
FROM PATIENT_INFO;

ExportSecret
SELECT SECRET_TBL.* INTO [dBASE 5.0;Database=C:\Diary\Files2Export].Wsecret
FROM SECRET_TBL;

ExportTension
SELECT TENSION.* INTO [dBASE 5.0;Database=C:\Diary\Files2Export].Wtension
FROM TENSION;

ExportUnknown
SELECT UNKNOWN.* INTO
[dBASE5.0;Database=C:\Diary\Files2Export].Wunknown
FROM [UNKNOWN];

ImportAllha
INSERT INTO ALLHA
SELECT Lallha.*
FROM Lallha;

ImportContact
INSERT INTO CONTACT_MASTER
SELECT Lcontact.*
FROM Lcontact;

ImportCore30
INSERT INTO CORE30
SELECT Lcore30.*
FROM Lcore30;

ImportMainData
INSERT INTO MAINDATA_MASTER
SELECT Lmaindat.*
FROM Lmaindat;

ImportMigraine
INSERT INTO MIGRAINE