1984

Resort condominium reservation system

Warren M. Bartlett

The University of Montana

Let us know how access to this document benefits you.
Follow this and additional works at: https://scholarworks.umt.edu/etd

Recommended Citation
https://scholarworks.umt.edu/etd/3413

This Thesis is brought to you for free and open access by the Graduate School at ScholarWorks at University of Montana. It has been accepted for inclusion in Graduate Student Theses, Dissertations, & Professional Papers by an authorized administrator of ScholarWorks at University of Montana. For more information, please contact scholarworks@mso.umt.edu.
A Resort Condominium Reservation System

by

Warren M. Bartlett

M.B.A. University of Montana, 1981

Presented in partial fulfillment for the degree of

Master of Science

UNIVERSITY OF MONTANA

1984

Approved by:

Chairman, Board of Examiners

Dean, Graduate School

Date
Abstract

Bartlett, Warren M., M.S. August 1984 Computer Science

A Resort Condominium Reservation System

Director: Dr. John R. Barr

The purpose of this project was to 1) determine the Specifications and Requirements for a Reservation System for a Resort Condominium property, 2) to use these Specifications and Requirements to develop the Design for the software system, and 3) to use this Design to Implement the Software System. The methodology used in the Requirements and Specification stage was the Data Flow technique. This is represented with Data Flow Diagrams, Process Descriptions, and Data-Dictionary. The Design stage utilized structure charts and module descriptions to complete the design. The implemented system was developed on a microcomputer using a data base management system. The entire process was done on the microcomputer with assistance of only the tools currently available on this type of machine.

The major portion of the effort that was spent on the project fell in the first two phases. The Requirements and Specifications, in particular, took a substantial amount of time primarily due to the necessity of user involvement. This effort, in the early parts of the project, dramatically reduced the time necessary to implement the system. The coding into the Data Base Management System (DBMS), dBASE III, was easily completed in a short time with relatively few difficulties.
# Table of Contents

Abstract ......................................................... ii.
Table of Contents .............................................. iii.
List of Illustrations ........................................... iv.
Chapter 1 - Introduction .............................. 1.
Chapter 2 - Existing System .......................... 6.
Chapter 3 - The Proposed System .................. 14.
Chapter 4 - The Design Structure Charts ...... 21.
Chapter 5 - The Design Module Descriptions .... 32.
Chapter 6 - The Implementation .................... 36.
Chapter 7 - Conclusion ................................. 42.
Appendix A - Existing System Data Flows .... 44.
Appendix B - Existing System Subroutines ..... 48.
Appendix C - Existing System Files ............... 52.
Appendix D - New System Data Flows ............. 56.
Appendix E - New System Data Dictionary ...... 64.
Appendix F - New System Files ....................... 72.
Appendix G - New System Process Descriptions .. 79.
Appendix H - New System Structure Charts .... 92.
Appendix I - Module Design ......................... 100.
Appendix J - Screen Designs ......................... 137.
Appendix K - Data Base Structure ................. 144.
Appendix L - Implementation ....................... 155.
A Selected Bibliography ................................. 218.

iii
List of Illustrations

Figure 3-1 New System Context Diagram........ 15.
Figure 4-1 Data Flow - Modify Reservation..... 24.
Figure 4-2 Data Flow - Generate Reports....... 25.
Figure 4-3 Structure - Modify Reservation..... 29.
Figure 4-4 Structure - Generate Reports....... 30.
Figure 5-1 Description - Modify Reservation... 32.
Figure 6-1 Implementation - Modify Reservation 39.
Chapter 1 Introduction

The resort condominiums are a relatively new entity in the Hospitality industry. They are physically designed and constructed as normal condominium complexes with some additional features and services normally associated with more conventional lodging properties such as hotels and motels.

These types of properties are typically owned by individual investors or investor groups but are operated by management companies in much the same manner as resort hotels. These condominiums configurations are NOT the so called "time-shared" condominiums in which blocks of days or weeks are owned by different people or groups. The individual purchases separate units like any other real estate except that they are usually bought as investments primarily for the tax benefits associated with the ownership of this type of property. The management company then rents the unit and returns a portion of the rental income to the investor as income. They are typically rented on a day-to-day or week-to-week basis. This requires that the organization and management of the properties be set up in much the same manner as a hotel. They must maintain the typical hotel organizational features like front desk, reservations, back office, maintenance, housekeeping, etc.

These similarities with hotels are the reasons that the type of information systems most often found in these
properties is structured and organized almost identically to their hotel counterparts. In fact, most systems in use are older hotel systems with very minor (if any) additions or changes. Usually, there has been no attempt to accommodate the somewhat unique features of the condominium hotel configuration.

The unique features associated with these properties are exactly what make the hotel systems somewhat "clumsy" to use in that environment. A typical feature of condominium construction and design is the ability to "lock-off" a unit. This enables a single, multiple bedroom unit to be divided into two or more individually rentable units. This type of unit is not typically found in hotels or motels. Thus, most of the information systems used in hotels are unable to handle the rental of the separate parts of the unit. The systems available in condominiums usually just ignore the additional parts of the unit and the result is lost revenues for the owners because the rest of the unit lies empty.

An additional complication associated with this type of property is multiple ownership. The complex is not owned by one person or organization as a hotel or motel usually is, but by independent owners or investor groups. The owners typically derive some income from the individual units that they own. The income is usually, but not always, based on the actual revenues generated by their units.
This implies that the system must be able to allocate the use of the rooms in a manner which treats all of the owners fairly. The assignment of the units cannot be determined in the same manner as hotels where the entire property and its generated income is allocated to a single owner.

These two problems alone have led to a general dissatisfaction with most of the existing automated systems in use by these kinds of properties. Many properties, including several communities with large centralized reservations, are using manual systems because of the inability of available software to adapt to hotel-style condominiums.

There have been a few attempts to build systems to accommodate the special needs of the condominium. Some properties and/or management companies have developed their own automated systems to handle their particular needs. Most of these are too specialized for the particular properties that they were designed to work with. The systems were just designed to handle the problems associated with the particular property and they lack the necessary flexibility to be of use to the industry in general.

This project is the development of a reservation system specifically designed for the small condominium resort. It will consist of the specifications and requirements of the system, the design of the system, the user manuals, and the implementation of the system on a microcomputer.
The requirements and specifications will be expressed with the technique developed by Tom DeMarco using Data Flow Diagrams.¹ This technique is built around three elements used to describe the system to be developed. These are:

1. Data Flow Diagrams used to describe how the data flows through the system.
2. Data Dictionary used to describe each data flow and the files in the system.
3. The mini-spec, a structured English description of the low level process bubbles in the Data Flow Diagrams.

The design phase will be accomplished using a technique similar to the Structured Design technique described by W. P. Stevens, et. al.² The actual design technique used is the structured design methodology as described by Kathleen Dolan.³ This technique is based on transforming the Data Flow Diagrams into structure charts, refining the structure charts, and then using primarily procedural methods to specify the function of all the modules in the structure chart. After the design is complete, the system will then be implemented on an IBM microcomputer. The entire development,

from requirements and specifications stage to the actual implementation will be accomplished on the microcomputer.
Chapter 2 Overview of an Existing System

In order to build the system, the first step is to examine the existing system. The existing information system described here is a system in use today in a hotel condominium in Denver, Colorado. This particular system is not the only one that has been examined. Several others have been examined and specific, unique features of these systems will also be mentioned. None of the systems will be described in detail because none of them handle the condominium environment very well. In addition, the software to be developed will be for a new organization called Hospitality Resources International, Suite 6000, Stapleton Plaza, Denver, Colorado, which has no existing system at this time.

The properties for which the existing system is used are not really resort condominiums. They are in the city of Denver itself and are generally used for an alternative to a hotel type of lodging. They do not have the additional services found in resort systems like food and beverage service. The complexes are simply typical condominium apartment complexes rented as short term units rather than long term monthly rentals.

However, the other services provided by the current system are fairly typical for lodging facilities. A reservations system must be maintained for the properties. The
front office services are provided for guests to check-in and out. The housekeeping and maintenance services are required and provided just as they are in hotels and resorts.

The existing automated system performs primarily the front-desk and reservations functions for the Denver organization. Some back-office accounting detail is maintained in the system although it is not very comprehensive. There is a maintenance and housekeeping interface into the system but it maintains no effective or useful historical data. The system does produce some useful management reports but does not provide very complete control over the data.

As can be seen from the Data Flow Diagrams in Appendix A, the core of the system is the reservation system. This is the primary way that information is entered into the organization. A reservationist takes a reservation call by phone and accepts the reservation data from a potential guest. This includes, among other things, the date and room requirements of the customer. With this information, the reservationist can determine if a room is available that satisfies the needs of the individual or organization. At this point the particular apartment is selected based on the data maintained in a unit-data file and the rest of the required information is collected from the customer.

At this point, a confirmation letter, which specifies all of the details of the particular reservation and the deposit requirement, is produced and sent to the customer.
The reservation details are then stored in a central file where they can be accessed easily for examination, alteration, and cancellation if the customer changes any of the details of their potential stay. The ability to examine reservations by name, unit, or date is provided by the system.

Based on the information maintained in the central file, various reports can be produced from the system. Usage reports which provide information on the usage and availability of the individual units are probably the most useful on a daily basis. Some marketing information can be extracted, such as companies using the system, their addresses and phone numbers for potential future use. Also a report describing the receivables, both deposits owed and payments due, is available.

The next step in this system is the check-in of the guest. This is a front desk function but is performed by the reservationists, since a formal front desk is not maintained in this particular property. This is due primarily to the fact that these properties are not really resorts or hotels and do not have a physical front desk. The reservationists perform both the functions of the front desk clerk and the back-office reservationist. At this point, very few additional pieces of data are entered into the system. Primarily, the guest provides some payment for deposit and prepayment of rental charges and this data is entered into the central file. Then, the reservationist gives
the guest the necessary materials like keys, receipts, regulations, etc. and directs the guest to their accommodations.

When the guest checks out, they do so, again, with a reservationist. The payment and return of keys is handled at this time. The payment detail is entered into the system and stored in the central file along with the check out date and time. The guest is provided with a receipt to complete the transaction.

The information system in use in this organization is already automated. In its current state, it consists of about 100 subroutines written in FORTRAN (see Appendix B for a list of the routines). These range in size from about seven lines of code to as large as several hundred lines. In total, the entire system is probably well over 20,000 lines of relatively uncommented, unstructured code. Thus, although this system has been designed for the condominium configuration, there are several problems with the system in its current state.

Probably the most serious problem stems from the nature of the development process for this software system. The developer was an owner of the organization and thus understood the needs of the organization fairly well, but had little formal training in software development. No study of the requirements of the organization was ever done. No design, general or detailed, was ever completed. None of the tools
of structured coding were ever used in the implementation.

The amount of external documentation for the system could be put on about five pages and the internal documentation is minimal, at best! The system has simply evolved into the system that it is today.

Due to the lack of any engineering, the software has reached the point where it is practically impossible to maintain. Most of the subroutines are NOT functionally independent. They are strongly coupled together through COMMON data areas used in FORTRAN. Coupling can cause serious problems, particularly during the implementation and maintenance phases of the software's life. A Small change in one module can cause unexpected changes in other parts of the system.

An example of the kinds of problems encountered with the existing system occurred when the tax rate changed in the state of Colorado. The effort necessary to alter the system to conform to the new tax schedule was substantially more than it should have been. When the process was complete, it was found that the tax rate had been computed incorrectly all along! Fourteen subroutines had to be modified before the system was handling the new tax rate correctly. The

---

4Coupling is a measure of the strength of association established by a connection from one module to another. Strong coupling complicates a system since a module is harder to understand, change, or correct by itself if it is highly interrelated with other modules.
tax rate, the computed numerical value, occurred in ten modules; the tax was actually computed in ten separate, different places in the program!

This was not the only problem found in the software. There are parts of the system that have never functioned correctly. Because of the interaction between modules, the system almost appeared to behave in a nondeterministic manner. There is a substantial amount of redundant code other than the tax rate problem described above. There are three separate modules which calculate a guest's bill. All of the modules were not calculating the amount in the same manner. The system would compute the amount on the final bill differently than it would compute the amount on the accounts receivable report for the management. The reports gave the impression that the customer owed the organization money when, in fact, the bill was paid or worse, sometimes it appeared that the customer had paid their bill in full when they actually owed the organization substantial sums of money!

The system makes little distinction between the data for an active reservation, a current guest, an inactive reservation, or previous guest. All of the data associated with each of these different groups is maintained in the same central file. This file has a tendency to get VERY large. The system had a utility routine to purge the old records from the file, but it had never worked. It had
a tendency to purge current records from the file! Due to constantly changing corporate policies, lack of confidence in the software, and a large amount of unused disk space, the solution to this problem was to keep ALL of the past data in the file.

This file also contained information that was never used or needed in the organization. The size of each record was also quite large, 1024 characters per record in the reservations file alone. With a typical size of 1000 records, this file was a megabyte in size. The file is described in Appendix C.

The unit file in the system maintains all of the information associated with any particular unit. Since this organization managed several different properties, the system grouped all of the units for all of the properties into this single central file. This caused the unit file to be quite large. This file's contents are also described in Appendix C.

The software had to be able to handle all the different characteristics of the properties. This also made it more complex than necessary. It contributed to maintenance problems because the file had to be enlarged almost each time an additional property was added to the system. Of course, due to the lack of functional independence, this involved changing several modules that each opened and read from this file!
The evolutionary growth of the existing system and the lack of engineering of the product have resulted in an unreliable, unmaintainable product. At this stage in its lifecycle, the system is almost worthless. The complexity of the system has increased to the point that it is difficult and costly, if not impossible, to maintain it. The entire system needs to be redesigned and rewritten.
Chapter 3 The Proposed System

From the analysis of the existing system, it is apparent that it attempts to do too much. To develop a new system to accomplish all of the tasks and functions of the old system would be beyond the scope of this project and could result in the same type of product. The size of the project needed to be more manageable. The first step undertaken in the development of the new system was to break it down into smaller, relatively independent, systems. Then the decision on which part of the system to develop could be addressed.

Since the core of the system, as it exists now, is the reservations portion, its function is the most important part. This is the part where all the data enters their information system. In addition, this is the primary function that needs to be specialized for the unique problems associated with the condominium configuration.

In analyzing the overall system, it was found that the reservations system was reasonably independent of the other areas of the business. It interfaces to the customer, the accounting office, the front desk function, and, of course, the management. The interfaces are fairly well defined and fairly straightforward. This fact can be seen in the Context Diagram of the proposed system (Figure 3-1).
Context Diagram
Proposed System
Figure 3-1
This is a diagram which represents the data flows, their sources and destinations for the proposed reservations system. "The Context Diagram documents the domain of the study by showing the set of data flows that cross into and out of the domain".\textsuperscript{5}

The development of this diagram involved several stages. The first of which was the detailed study of the existing system's data flows. These flows are shown in the data flow diagrams of the existing system (Appendix A). From an examination of the physical data flow diagram, it is evident that the Reservation Department interfaces with the other parts of the organization primarily through the two files and some reports. Although the individual employees who perform the reservationist's functions may be the same as the front desk clerk, the jobs are actually relatively independent of each other, except for the interface through the two files. The data flow diagram displays this independence by separating them into two bubbles. This obvious separation of function and data flows, along with the other factors previously mentioned, lead to the decision to develop, from scratch, a totally new reservation system.

The next step was to examine what kinds of data should have been in the system and how it would flow through the

\textsuperscript{5}DeMarco, p. 76.
new system. This stage was completed with extensive assistance and input from the people at Hospitality Resources International. This user involvement was absolutely necessary to complete the specifications of the new system.

The structured system specification began with the top-level data flow diagram for the proposed system. This diagram shows the major data flows and their sources and destinations in the new system (Appendix D). The current version of the top-level picture has evolved through about eighteen to twenty previous versions. Every time the diagram was finished, a careful examination and analysis of the data flow representations and the "bubbles" would reveal flaws in the specification. It would then be done again and reevaluated. This iterative, refining process was necessary to arrive at a complete specification. The complexity of the system required this kind of technique to ensure that the system was correctly described and none of the details were missed or forgotten.

Once this top level diagram was constructed, the leveling process began. This step involved the breakdown of each of the "bubbles" in the top level data flow diagram. This was done to make the system's complexity more understandable and manageable. Each process in the top level diagram represents a large amount of transformation effort to change the data inflows to the data outflows. To show this trans-
formation, each of the processes is represented by its own diagram (see also Appendix D).

Thus, process number one, Accept New Reservation, is described by its Data Flow Diagram. This diagram shows the flow of data and the processes that transform the data in this single process of accepting a new reservation into the system. This diagram's process bubbles are labeled or named with the description of what occurs in the process, just as the top level diagram. The bubbles are numbered with 1.x to indicate that they are part of the top level diagram number 1.

The data flows in the Accept New Reservation diagram are all labeled. The flows which enter the diagram and leave the diagram correspond to the flows entering and leaving the Accept New Reservation process bubble in the top level diagram. These data flows and the data flows in all of the diagrams are described in the Data Dictionary for the new system in Appendix E.

The Data Dictionary has a description of all of the data flows in the diagrams for the new system and their component parts. Each data flow can be composed of several items and the dictionary shows each flow's composition. The dictionary is in alphabetical order to assist in finding the description of any component in the diagrams.

The Data Flow Diagrams also contain files which represent the areas where data is stored. Most information systems
need to recall stored data and the arrows on the diagrams show that data is both stored and retrieved from these files. These files must also be described so the specifications include the elements needed later in the design of the system. The files and their contents are described in Appendix F.

The last elements in the Data Flow Diagrams that need to be described are the process bubbles. These are the elements in the diagrams which represent those places where the data flows go through a transformation of some kind. The description of these processes represents what DeMarco calls the minispec. The minispec specifies "...the relationships that apply among the data flows." All of the processes in the lowest level data flow diagrams are described in Appendix G using the technique called Structured English.

The structured English is a type of "pseudocode" that represents the process that occurs in the "bubble". These descriptions are in a human readable form. In fact, they can, in general, be read and understood by individuals who know nothing about computers and programming languages. This is deliberate. The specifications must be developed with the assistance of the users of the system. They, the users, must be able to understand the specifications.

---

6 DeMarco, p.62.
7 Ibid, pp. 169-213.
to determine if they are correct. The process descriptions provide this capability.
Chapter 4 The Design Structure Charts

With the completion of the specifications in the Data Flow diagrams, Data Dictionary, and Mini-Spec, the next stage was the design of the system. At first glance, the initial description of the system with the Data-Flow technique appears to provide enough information to actually proceed to the coding stage of the development process. This method of providing specifications for a system does assist in the design. However, a closer look at the specifications reveals that there are several critical items missing from the specifications that are necessary to actually produce the code for the system.

There are three basic reasons why the code cannot be produced directly from the Data Flow diagrams. First, The Data Flow diagrams do not provide any kind of control information. All of the processes in the diagrams appear to be independent of each other with the exception of the data that actually flows from one to another. In reality, programs do control each other. One process often determines whether or not another process is activated. Since this control is not represented in the Data Flow diagrams and the related documents, we must provide the control information in an intermediate step, before the code. This is accomplished in the Design stage.

---

Dolan, p. 50.
The second problem with using the Data Flow diagrams to produce code is that the diagrams and the technique actually ignore most processing for exception or error conditions. Again, this type of processing is absolutely necessary in any kind of "real world" programs, especially when they interface with people. So our design must include the development of the error handling processing.

The third major problem with the Data Flow diagrams is that they assume that a never-ending flow of input data exists. They do not provide any insight into what processing is necessary at the end of the input stream. The details of the necessary steps for the end of input condition must be described in the design stages.

The design methodology used in this project is based on the concepts of structured design and the structure chart. A structure chart is really a visual representation of the program. "It shows the overall function of the program, as well as details of each of the programs subfunctions."9 The structure chart can be derived directly from the data flow diagrams.

There are two different techniques used to derive the structure charts from the data flow diagrams. Both of these techniques were used in the design of the reservation system. The first of these is called transform analysis.

9Dolan p.45.
In this method, the data flow diagram is sectioned into three parts. The first is the afferent or input branch where the data is collected and put into a form ready for processing. The second is the actual processing portion of the diagram that performs the basic function of the system. The last branch is the efferent or output branch which "... formats and disperses the output."¹⁰

An example of the use of transform analysis in this project is in the module that modifies existing reservations. The data flow diagram (Figure 4-1) was examined to determine where in the flow the input flows were actually transformed into output flows. This was done to identify the area of central transform. Then, the initial structure chart is directly derived from the data flow diagram by drawing this area as the top of the structure chart. Some minor adjustments to the names of the process bubbles and the inclusion of read and write modules gives the first cut structure chart under this method.

An alternative technique called transaction analysis was also used to generate first-cut structure charts in this project. Figure 4-2 shows the data flow diagram for the reporting function of the reservation system. This diagram displays a process that is obviously case structur-

¹⁰ Jensen and Tonies, p.186.
Data Flow Diagram
Modify Existing Reservation
Figure 4-1
Data Flow Diagram
Generate Reports
Figure 4-2
ed. That is the transaction can involve one of several possible processing bubbles. In this example, there can be one of several different type of reports produced. This involves a choice. The transaction type (or choice) must be determined and then, based on the type, the particular report can be then produced by the system. This type of system typically has a transaction center which controls the invocation of any of the other subsystems. This bubble is placed at the top of the first cut structure chart and the design process begins.

Both of these techniques generate first cut structure charts. As the name implies, these structure charts are not at a stage where the implementation can begin. There must now come a stage where these charts are analyzed and refined to produce a good design. The primary concept used in this analysis is the engineering idea of the black box.

A black box in software design is a process that performs some function. It is not necessary to know the details of how the process actually performs the function. The process that uses a black box module must know only what the function of the module is and how to use it. It does not need to be concerned with how the process was actually implemented or how it performs the function.\textsuperscript{11}

\textsuperscript{11}Jensen and Tonies, P. 16.
The first cut chart is just the starting point. The modules in this chart are not finished and are really just rearranged data flow diagram process bubbles. Each module or box in the structure chart must be examined to determine if they satisfy certain criteria that make them black boxes. The characteristics of these black boxes make them especially attractive in software design. K. Dolan\textsuperscript{12}
describes these characteristics as follows:

1. Modules can be used without having to be understood internally.

2. Modules are independent of each other, except where they interface with each other.

3. They can be further subdivided into smaller modules, thereby reducing complexity by hiding details.

4. Program problems can be easily traced to a faulty module that can be diagnosed, replaced, or repaired.

5. They are easily understood.

An analysis of the cohesion of a module is one place to begin the refinement of the structure chart. Each module must contain all the necessary things that it needs to perform its function. This should be a single function. If the module performs more than one task, then it is a good candidate for a cohesion problem.

Coupling between modules is the amount of dependence that modules have on one another. If modules are highly

\textsuperscript{12}Dolan, p. 89.
coupled, a problem in one module can cause rippling effects throughout the system. If it is necessary to change the implementation of a module and it is in a highly coupled system, a simple change can involve altering many other modules to adjust for the change. Only the minimum amount of data should flow from one module to another. The interfaces between the modules must be kept to a minimum to reduce coupling.

Other factors must be examined to come up with a good design. The size of the modules should be small enough to minimize their complexity. One individual module should not control too many other modules. This also reduces the complexity of the controlling module. The usefulness of the module can also be evaluated. This technique can help to produce modules that can be used by several other processes. This reduces the complexity of the system by making changes easier to implement. If a file read is always done by a single module, then when the file structure is changed, only that one module must be changed.

After the analysis, the resulting structure chart should have been refined to reflect the actual structure of the program. Shown in Figure 4-3 is the refined structure chart for modifying existing reservations. The data flow diagram for this module was shown in Figure 4-1. The structure chart for the reporting function is shown in Figure 4-4. Note that this particular chart readily displays the case...
Structure Chart
Modify Existing Reservations
Figure 4-3
structure or choice structure that was evident in the data flow diagram shown in figure 4-2. The remaining structure charts for the new reservation system are contained in Appendix H.
Chapter 5 The Design Module Descriptions

Once the structure charts for the entire system had been constructed, the next step was the specification of each of the modules in the charts. There are several different techniques for specifying the modules in the design. One technique that is often used is called the procedural technique in which the module is specified through a written description of the algorithm the module must use to accomplish its function. Another technique used to describe the modules is called nonprocedural. With this method, the process is described through decision tables or layouts of forms and reports with, when necessary, a very brief description of the algorithm used by the module.

The technique used in this reservation project was primarily procedural in its orientation. In this system, the modules are described through a pseudocode or structured English description. For each of the primary modules in the structure charts, there is a description of the task or function provided by that module. An example is seen in Figure 5-1. This is the module description for the routine which allows modification of the existing reservations. As can be seen, the pseudocode almost appears as a programming language. This facilitates the actual coding since the program flows directly from the description.
DO Get Reservation Key (RETURNS Number or Name)

IF NO KEY
RETURN
ELSE
DO Get Current Reservation WITH Reservation Key
RETURNS Reservation Record
IF No reservation
RETURN
ELSE
DO Display Reservation WITH Reservation Record
Prompt user for Cancellation or Change
IF user does NOT want a cancellation
DO Change Reservation WITH Reservation Record
ELSE
DO Cancel Reservation WITH Reservation Record
ENDIF
DO Generate Confirmation WITH
Message Type and Confirmation Data
ENDIF
ENDIF
RETURN
The English-like description of the module can be compared to the structure chart (Figure 4-3) and it can be seen that additional information is provided by the description. The algorithm for this module is easily explained with the procedural description. The description goes beyond the structure chart and bridges the gap between the chart and the next step, the actual implementation.

Some additional nonprocedural techniques were also used to describe other features and characteristics of the system. Examples of these can be seen in the headings of each of the pseudocode descriptions where some details are specified. In addition, the screen designs and database structures for the system were separately specified.

The screen designs are contained in Appendix J. They provide a detailed description of the actual appearance of the screens that the user will face. These descriptions are produced in order to provide a consistency in the user interface. The data input and data editing portions of the new system use the same screens for this very purpose. The menus used in the system are also described in this Appendix. The design should include this information if consistency is necessary.

The actual structures of the files (or databases) used in the system are contained in Appendix K. This additional information is provided in the design phase to keep the
data storage consistent. These descriptions provide field names and sizes for all of the files. During the implementation of the various modules that use this data, it is necessary to have this information.
Chapter 6 The Implementation

Once the design had been completed, the implementation of the system could begin. At this point, the decision had to be made concerning what language would be used for the system implementation. Several items were considered before the final decision was made.

During the initial work on this project it was decided that one of the factors to consider in choosing the implementation language had to be transportability. Ideally, the software would not be dependent on any hardware or any particular manufacturer. The range of machines considered was minicomputers and microcomputers. In this range of machines, one language is fairly transportable across most of the hardware. That language is C, the programming language usually associated with the UNIX operating system.

In terms of transportability, this language has distinct advantages. It is available on most small machines and it is fairly standard in all of its implementations. However, C is a relatively low level programming language and thus could possibly contribute to a longer development time.

Another factor considered was the reservation system itself. This system was a fairly typical business system. It had all of the usual characteristics associated with these systems. The system had to allow the entry and storage of data. It had to allow the alteration or modification
of the stored data and the access to the data had to be fairly flexible. Finally, it had to allow the reporting of the data stored by the system. These type of systems can typically be implemented much easier using a higher level language than C.

Two other recent factors helped to influence the ultimate decision. First, a new, enhanced version of a very popular Data Base Management System (DBMS) just became available. This was the software package called dBASE III from Ashton Tate of Culver City, California. This system was a complete rewrite of the most popular DBMS on microcomputers called dBASE II.

Although this system is not available on minicomputers, most hardware and operating systems have some sort of DBMS available for them. Also, dBASE III had been written in C to increase its transportability. It is speculated that it will soon be available on minicomputers using the UNIX operating system. These factors may ultimately make dBASE III very transportable.

The second factor to consider was the recent announcement of tools that allow the translation of dBASE II and dBASE III syntax directly to the programming language C. Thus, using this type of tool, the system developed in dBASE III could be translated to C and transported to most machines.
A brief description of the most important features of this package follows. dBASE III is a true relational DBMS which allows the creation, maintenance, and query of data files called relations. It has a built-in very high level programming language which provides all of the constructs necessary for implementing a software system using structured programming control structures.\textsuperscript{13}

It provides the sequence structure with procedure calls (DO) and RETURN. In addition it has the ability to pass parameters to the called routine and also has local (PRIVATE) variables implemented. Both the selection structures of the IF...THEN...ELSE...ENDIF and CASE...ENDCASE are provided. The iteration structure is implemented through the DO WHILE...ENDDO construct. In addition, the exit structures described by Jensen, et. al.\textsuperscript{14} are implemented. The escape construct is done through the EXIT statement which exits an active WHILE loop. The cycle structure is implemented through the LOOP statement which causes an unconditional branch to the beginning of a WHILE loop. The language does not have a GO TO implemented. In addition to these structured programming statements, the language has very sophisticated file handling and record locating


\textsuperscript{14}Ibid. pp. 261-263.
commands which simplify the implementation of business type systems.

Based on the reasons described above, the decision was made to implement the system using dBASE III. The powerful, structured programming language allowed an almost direct translation of the design module specifications to the implementation. An example can be seen by examining the procedural specification of the module designed to modify an existing reservation shown in Figure 5-1 and comparing that to the dBASE III implementation shown in Figure 6-1. It is very evident that the implementation of the module even reads almost exactly as the design. The minor differences are simply the addition of variables and a few statements designed to enhance the user interface.

The programming language associated with the dBASE package provides all the tools to implement the application using the software engineering tools of structured coding. In addition, it provides very high level statements for locating records and data in the files maintained by the system. This makes for an easy extension of the design into code. The file structure described in Appendix K are exactly what is implemented in the dBASE III system. In fact the whole Appendix was simply printed by the DBMS itself.

The implementation proceeded well with the worst difficulty being the lack of experience on the particular package.
* Program...: Modify Existing Reservation (MODIFRES)
* Author...: Warren M. Bartlett
* Called by: MENU
* PARAMETERS None
PRIVATE answer
DO WHILE .T.
  number = " "
  name = " "
  DO getresky WITH number, name
  IF name = " 
    IF number = " 
      RETURN
    ELSE
      USE reservat INDEX resnum
      FIND &number
    ENDIF
  ELSE
    USE reservat INDEX resname
    FIND &name
  ENDIF
  IF EOF()
    answer = " 
    @ 22,2 SAY "Reservation does NOT exist... Press Enter
    @ 22,65 GET answer
    READ
    ELSE
      DO copyvars
      DO dispres
      answer = " 
      @ 24,3 SAY "<E>dit or <C>ancel (Enter to return)?"
      @ 24,55 GET answer PICTURE ":!"
      READ
      IF answer = "E"
        DO editres
      ELSE
        IF answer = "C"
          DO rescanc1
        ELSE
          EXIT
        ENDIF
      ENDIF
    ENDIF
  ENDDO
USE
RETURN

Module Implementation
Modify Existing Reservation
Figure 6-1
The DBMS does provide some development tools to assist in application development. These include a built-in full screen editor which provides the ability to modify the programs without leaving the DBMS. Also, included in the system is a programming tool to generate programs for the screens. This simplified the generation of the programs to produce the screens that are described in Appendix J.

The testing of the system was done during the implementation stage. This was accomplished through the use of Top-Down and Incremental Testing as described by Yourdon and Constantine.15 This technique uses "stubs" which allow top level modules to be tested prior to the implementation of the bottom level modules. The biggest advantage of this technique is that it allows the testing of one new module at a time. This technique was essential for this project because of the lack of familiarity with dBASE III and its programming language.

The system does provide some tools for the testing of programs, although a symbolic debugger would have been helpful. The programs can be run on a single step mode and the commands or statements echoed as they are executed. These tools were used all during the implementation stages.

---

Chapter 7 Conclusion

At this point the software exists. The contribution of the preliminary work to the overall project was substantial.

The Requirements and Specification stage was difficult to complete due to two factors. First, the user was in Denver during most of the work. Telephone and mail communications were used to get assistance during this part of the project. It would have definitely been easier if this work had taken place closer to the people who are knowledgeable in the industry details. The results were acceptable but it would have taken far less time with closer user involvement.

The second problem was caused by the fact that the task was completed on a microcomputer. Several automated tools, in particular, a dictionary manager for the data dictionary, would have helped considerably during this phase.

The design stage took a substantial amount of time. This could have been caused by the communications problem just described. Some of the details in the design effort had to be discussed with the user. Once again, the communications problem slowed this review process down considerably. Slowing this process may not have been all that detrimental to the ultimate result. This slower design process may have contributed to a higher quality in the final design.
The results of the design effort were very evident when the Implementation stage began. The coding process was completed very quickly. Although the design was kept away from hardware and software considerations, it progressed quite smoothly. Another factor that contributed to the ease of implementation was the Data Base Management System, dBASE III. With the ability to use the basic elements of structured coding, the design to code phase was relatively trouble free.

The primary cause of bugs in the software was not due to design errors. It was caused, for the most part, by a lack of understanding of the dBASE III package itself. Since this was the author's first major project using this package, the overall results, in both quality and productivity, were acceptable.

The ultimate measure of the quality of both the product and the development process, will come during the maintenance phase. The actual use of the system will determine the effectiveness of this engineering process. Unfortunately, this cannot be determined at this time.
Appendix A

Data Flow Diagrams

for the

Existing System
Data Flow Diagram
Existing System
Data Flow Diagram
Existing System - Reservation Department
Data Flow Diagram
Existing System - Check-In
Appendix B
Subroutine List
For
Existing System
Utilities:

ADRSQU : Remove "Non-Essential" Chars
DAA : Input A Date
DAIN1 : Calculates Integer Day from Date
DAIN2 : Calculates Date from Integer Day
LAL20 : Change Character Sizes on LAL20
POS : Positions Cursor and...
RED : Positions Cursor and Reads Value Range
SCREEN : Displays Screen from File "SCREEN.DAT"
TIMER : Provides Delay

System Modules:

6 : Opens/Closes Company Address Records File
7 : Calcs/Prints Invoices
9 : Drives 7
10 : Opens Files 40,41
11 : Squeezes File DCRF2 Into Date Order
12 : Determine Next Reservation Number
13 : Gets Location of Reservation from Number
15 : Calc Number of Days Payment is Due for Crib
16 : Calc WEIGHTING Factor (?)
17 : Calc Rates and Dates for Stay
18 : Changes (?)
19 : Put "Pause..." Message up for Printer Set-up
20 : Determine Location of Unit Data given Unit
21 : Packs Unit Address File
22 : Put Unit Info in Lower RH Corner of Screen
23 : Read into Array A (Reservation Info Screen)
24 : Read into Array A (Reservation Info Screen)
25 : Read into Array A (More Reservation Info)
26 : Prints Summary of Charges Due at Check-In
27 : Checks for Overlap in Reservations
28 : Display Unit Info on CRT
29 : Modify Unit Data
30 : Adds New Unit
31 : Opens and Fills File 46
32 : Puts "MINI" Avail List on CRT
33 : Display Reservations
34 : Display Reservation Info
35 : Display Reservation Info
36 : Display Reservation Info
37 : Change Reservation Info (Screen 22)
38 : Change Reservation Info (Screen 31)
39 : Change Reservation Info (Screen 45)
Reservations
Unit Data
Print Availability List
Housekeeping
Account Status
Payment Status for "B" Account on CRT
General Subroutine (EMPTY)
Print Invoices for "C" File
List Reservations by Company Name
Enter Comment OR Verify Employment
Waiting List
Print Reservation Confirmation
Front Desk Program (cont. 60)
Delete OR Transfer Reservation
Prints ALL (?) File Data
Display OR Change Comments
Display Check-Ins
Prints Reservation Confirmations
Enter Housekeeping Holds
Front Desk Program (cont. from 53)
Weekly Guest List by Complex
Hunters Guests for Pool Towels (?)
Returns Number of FREE Lines in Res. File
Maintence Program
Inventory of Cribs, Roll-A-Ways, Hi-Chairs
Starts ALL Programs (First CALL from DCRM!)
Print Occupancy Graph
Calculates Payment Due
ADD, DEL, or PRINT Company Names
Return Rates (nightly, maid)
Part of Invoice Print (cont. from 79)
Housekeeping Schedule
Phone List / Day for REG Cleaning (?)
Enter Info In Payment File
Displays Check-Outs
Distribute to Payment File
Invoice MAIN LINE
Calculates Payment Due
Prints Invoice for "A" or "B" (cont. 71)
Determines Number of LF's to Top-of-Page
Distribute to Payment File (?) (76?)
Date Paid Through
Transfer Reservation to New Reservation
Print Guests Elig. for Deposit Refunds
Print ALL File Data for a given Date
Print Invoice for Rate Plans "B" and "C"
Delete Old Reservations
Calculate Guest Payment Due
Print Reservation File
Marketing Program MAIN LINE
(?) Does the Letter Match?
51

93 : Displays Company Name and Address on CRT
94 : Print Housekeeping Schedule
95 : Weighting Factor for Hunter's Units (?)
96 : Statistical Info on Stored Reservations
97 : Statistical Info on Stored Res (Daily Basis)
98 : Status of Unit Cleaning
99 : Mailing Labels for Corporate Clients
Appendix C

File Structure
for the
Existing System
Guest File for Existing System
(NOTE: 2 Characters per Integer)

<table>
<thead>
<tr>
<th>INTEGER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RESERVATION NUMBER</td>
</tr>
<tr>
<td>2</td>
<td>UNIT NUMBER (1-300) = LINE NUMBER IN FILE</td>
</tr>
<tr>
<td>3</td>
<td>INITIAL FIRST NIGHT</td>
</tr>
<tr>
<td>4</td>
<td>INITIAL LAST NIGHT</td>
</tr>
<tr>
<td>5</td>
<td>DATE INITIAL RESERVATION MADE</td>
</tr>
<tr>
<td>6</td>
<td>RESERVATION INITIALS</td>
</tr>
<tr>
<td>7</td>
<td># CHILDREN UNDER 16</td>
</tr>
<tr>
<td>8</td>
<td># PETS</td>
</tr>
<tr>
<td>9</td>
<td># CRIBS</td>
</tr>
<tr>
<td>10</td>
<td>DATE TO DELIVER CRIB</td>
</tr>
<tr>
<td>11</td>
<td>DATE TO PICK UP CRIB</td>
</tr>
<tr>
<td>12</td>
<td># HI CHAIRS</td>
</tr>
<tr>
<td>13</td>
<td>DATE TO DELIVER HI CHAIR</td>
</tr>
<tr>
<td>14</td>
<td>DATE TO PICK UP HI CHAIR</td>
</tr>
<tr>
<td>15</td>
<td># ROLL A WAYS</td>
</tr>
<tr>
<td>16</td>
<td>DATE TO DELIVER</td>
</tr>
<tr>
<td>17</td>
<td>DATE TO PICK UP</td>
</tr>
<tr>
<td>18</td>
<td>AMT OF DEPOSIT GUEST WAS TOLD DUE AT CHK-IN</td>
</tr>
<tr>
<td>19</td>
<td>RATE PLAN UNDER A,B,C</td>
</tr>
<tr>
<td>20-24</td>
<td>RESERVATION NAME 5A2</td>
</tr>
<tr>
<td>25-29</td>
<td>FIRST NAME 5A2</td>
</tr>
<tr>
<td>30</td>
<td>GUEST DESIGNATION CODE (VIP ETC)</td>
</tr>
<tr>
<td>31</td>
<td>CHECK-IN LOCATION</td>
</tr>
<tr>
<td>32</td>
<td>MAID SERVICE SCHEDULE MADE AT CHECK-IN</td>
</tr>
<tr>
<td>33-35</td>
<td>HOME PHONE OF GUEST AC+##</td>
</tr>
<tr>
<td>36-39</td>
<td>BUSINESS PHONE OF GUEST AC##EXT</td>
</tr>
<tr>
<td>40-45</td>
<td>PERSON MAKING RESV IF OTHER THAN GUEST</td>
</tr>
<tr>
<td>46-48</td>
<td>PHONE NUMBER OF ABOVE PERSON</td>
</tr>
<tr>
<td>49-56</td>
<td>COMPANY NAME 8A2</td>
</tr>
<tr>
<td>57</td>
<td>HEARD ABOUT US THRU...</td>
</tr>
<tr>
<td>58-87</td>
<td>COMMENT RE INIT RESV OR CHECK-IN ONE LINE</td>
</tr>
<tr>
<td>88-127</td>
<td>MAIL CONFIRMATION TO</td>
</tr>
<tr>
<td>128-167</td>
<td>MAIL INVOICES TO AFTER CHECK-IN</td>
</tr>
<tr>
<td>168</td>
<td>DATE EMPLOYMENT VERIFIED</td>
</tr>
<tr>
<td>169</td>
<td>INITIALS OF PERSON THAT VERIFIED</td>
</tr>
<tr>
<td>170-174</td>
<td>NAME OF PERSON CONTACTED</td>
</tr>
<tr>
<td>175</td>
<td>FIRST INITIAL OF GUEST</td>
</tr>
<tr>
<td>176</td>
<td>DATE OF CHECK-IN</td>
</tr>
<tr>
<td>177</td>
<td>TIME OF CHECK-IN</td>
</tr>
<tr>
<td>178</td>
<td>TAX OVERRIDE (Y=NO TAX LESS THAN 30 DAYS)</td>
</tr>
<tr>
<td>179</td>
<td>NUMBER OF KEYS GIVEN TO GUEST</td>
</tr>
<tr>
<td>180</td>
<td>DATE KEYS GIVEN TO GUEST</td>
</tr>
<tr>
<td>181</td>
<td>INITIALS OF PERSON CHECKING GUEST IN</td>
</tr>
<tr>
<td>183</td>
<td>AMT OF SEC DEP RECD</td>
</tr>
<tr>
<td>Column</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>184</td>
<td>DATE RECD</td>
</tr>
<tr>
<td>185</td>
<td>METHOD OF PYMT</td>
</tr>
<tr>
<td>186</td>
<td>AMOUNT RETURNED</td>
</tr>
<tr>
<td>187</td>
<td>DATE RETURNED</td>
</tr>
<tr>
<td>188</td>
<td>METHOD OF RETURN (CC=CO CHECK, VI=VISA ETC)</td>
</tr>
<tr>
<td>189</td>
<td>CURRENT RESERVATION START DATE</td>
</tr>
<tr>
<td>190</td>
<td>CURRENT LAST DAY</td>
</tr>
<tr>
<td>191</td>
<td>CENTS*100 FOR SECURITY DEPOSIT RETURNED</td>
</tr>
<tr>
<td>192</td>
<td>NIGHTLY RATE FOR CONDO</td>
</tr>
<tr>
<td>194-195</td>
<td>LAST 4 DIGITS OF CREDIT CARD NUMBER</td>
</tr>
<tr>
<td>196</td>
<td>IF PLAN A, NIGHTLY RATE FOR SPL CLEANING</td>
</tr>
<tr>
<td>197</td>
<td>IF PLAN B, 30 DAY RATE</td>
</tr>
<tr>
<td>198</td>
<td>IF PLAN C, MONTHLY RATE</td>
</tr>
<tr>
<td>199-246</td>
<td>N(1,J)=DATE CHANGE MADE</td>
</tr>
<tr>
<td></td>
<td>N(2,J)=RESV INITIALS</td>
</tr>
<tr>
<td></td>
<td>N(3,J)=NEW START DATE</td>
</tr>
<tr>
<td></td>
<td>N(4,J)=NEW LAST DAY</td>
</tr>
<tr>
<td></td>
<td>N(5,J)=NEW MAID SERVICE SCHEDULE PLAN</td>
</tr>
<tr>
<td></td>
<td>N(6,J)=NEW NIGHTLY RATE FOR CONDO</td>
</tr>
<tr>
<td></td>
<td>N(7,J)=NEW NIGHTLY RATE FOR SPL CLEANING</td>
</tr>
<tr>
<td></td>
<td>N(8,J)=DATE RATE CHANGE IS EFFECTIVE</td>
</tr>
<tr>
<td></td>
<td>N(9-12,J)=NAME OF PERSON MAKING CHANGE</td>
</tr>
<tr>
<td>247</td>
<td>OVERRIDE ON DAYS TO BILL IN ADVANCE</td>
</tr>
<tr>
<td>248</td>
<td>CREDIT CARD TYPE</td>
</tr>
<tr>
<td>249</td>
<td>EXPIRATION DATE</td>
</tr>
<tr>
<td>250-255</td>
<td>CC NUMBER 6A2</td>
</tr>
<tr>
<td>256-261</td>
<td>REFERRAL NAME 6A2</td>
</tr>
<tr>
<td>262-277</td>
<td>N(1,J)=DATE OF COMMENT</td>
</tr>
<tr>
<td></td>
<td>N(2,J)=INITIALS OF PERSON INPUTTING COMMENT</td>
</tr>
<tr>
<td>278-312</td>
<td>UNUSED</td>
</tr>
<tr>
<td>313-341</td>
<td>N(1,J)=DATE OF SPECIAL CHARGE</td>
</tr>
<tr>
<td></td>
<td>N(2-5,J)=DESCRIPTION OF CHARGE 4A2</td>
</tr>
<tr>
<td>342</td>
<td>INVOICE INCREMENT NUMBER START AT 10</td>
</tr>
<tr>
<td>343-432</td>
<td>N(1,J)=DATE OF PAYMENT</td>
</tr>
<tr>
<td></td>
<td>N(2,J)=METHOD OF PAYMENT</td>
</tr>
<tr>
<td></td>
<td>N(3,J)=PURPOSE (SD=SEC DEP, RE=RENT, PH=PHONE)</td>
</tr>
<tr>
<td>433-434</td>
<td>DATES FOR SPECIAL CREDITS/COMMENT FOR DESCRIPT</td>
</tr>
<tr>
<td>435</td>
<td>DATE CLEANED AFTER THIS GUEST CHECKED OUT</td>
</tr>
<tr>
<td>437-512</td>
<td>X(1-6)=SIX SPECIAL CHARGES</td>
</tr>
<tr>
<td></td>
<td>X(7-8)=TWO SPECIAL CREDITS</td>
</tr>
<tr>
<td></td>
<td>X(9-38)=THIRTY PAYMENT AMOUNTS</td>
</tr>
</tbody>
</table>
Unit File for Existing System
(Note: 2 Characters per Integer)

<table>
<thead>
<tr>
<th>INTEGER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>Unit Address IE 7337A</td>
</tr>
<tr>
<td>4-7</td>
<td>Complex Name</td>
</tr>
<tr>
<td>8-27</td>
<td>Actual Unit Address 2 Lines/20 Char Per</td>
</tr>
<tr>
<td>28-29</td>
<td>Unit Phone #</td>
</tr>
<tr>
<td>30</td>
<td>Day of Week for Normal Once/Week Housekeeping</td>
</tr>
<tr>
<td>31</td>
<td>#Bedrooms</td>
</tr>
<tr>
<td>32</td>
<td>#Baths (*10) IE 1.5 Baths = 15</td>
</tr>
<tr>
<td>33</td>
<td>Color Carpet</td>
</tr>
<tr>
<td>34</td>
<td>Basement (Y/N)</td>
</tr>
<tr>
<td>35</td>
<td>Washer/Dryer in Unit</td>
</tr>
<tr>
<td>36</td>
<td>Fireplace in Unit</td>
</tr>
<tr>
<td>37</td>
<td>Location Code</td>
</tr>
<tr>
<td>38</td>
<td>Date Unit First Avail</td>
</tr>
<tr>
<td>39</td>
<td>Date Unit Terminated</td>
</tr>
<tr>
<td>40</td>
<td>Garage for Unit</td>
</tr>
<tr>
<td>41</td>
<td>Rate Schedule Number</td>
</tr>
<tr>
<td>42</td>
<td>Children Allowed (Y/N)</td>
</tr>
<tr>
<td>43</td>
<td>Pets Allowed (Y/N)</td>
</tr>
<tr>
<td>44-56</td>
<td>Unit Description 13A2</td>
</tr>
<tr>
<td>57</td>
<td>Schedule A 4-29 Day Rate</td>
</tr>
<tr>
<td>58</td>
<td>Schedule A 30-59 Day Rate</td>
</tr>
<tr>
<td>59</td>
<td>Schedule A 60+ Day Rate</td>
</tr>
<tr>
<td>60</td>
<td>Schedule B 30 Day Rental Rate</td>
</tr>
<tr>
<td>61</td>
<td>Schedule C 3-5 Mo Rental Rate</td>
</tr>
<tr>
<td>62</td>
<td>Schedule C 6 Mo Rental Rate</td>
</tr>
<tr>
<td>63</td>
<td>Date of Last Phone Bill Processed for Unit</td>
</tr>
<tr>
<td>64</td>
<td>Unused</td>
</tr>
<tr>
<td>65</td>
<td>Sales Tax Rate IE 7.5% Stored as 75</td>
</tr>
<tr>
<td>66-128</td>
<td>Reservations Stored</td>
</tr>
</tbody>
</table>

N(1, J) = Resv #
N(2, J) = Start Date
N(3, J) = Last Day
Appendix D

Data Flow Diagrams

for the

New Reservation System
Data Flow Diagram
Proposed System - Top Level
Data Flow Diagram
Proposed System
Accept New Reservation - Process 1
Data Flow Diagram
Proposed System
Modify Reservation - Process 2
Data Flow Diagram
Proposed System
Generate Management Reports - Process 4
Data Flow Diagram
Proposed System
File Management - Process 5
Data Flow Diagram
Proposed System
Maintain Current Guests - Process 6
Appendix E

Data Dictionary for the Reservation System
ADDITIONAL FEATURES = Comment

ADDRESS = [Street Number + Street | PO Box Number]
+ City
+ State
+ Zip Code

ARRIVAL DATA = Date
+ Unit Number
+ Reservation Name
+ Nightly Rate
+ (Group Name)
+ (Arrival Time)
+ (Number of Cribs Needed)
+ (Number of Roll-A-Ways Needed)
+ (Number of High-Chairs Needed)
+ (Pets)

ARRIVAL DATE = Date

ARRIVAL DEPARTURE REPORT = Date
+ { Arrival Data }
+ { Departure Data }

ARRIVAL TIME = Time-Of-Day

BILLING INFORMATION = Name + Address

CANCELLATION APPROVAL = Initials

CANCELLATION DATA = Reservation Name
+ Reservation Number
+ Canceled Date
+ Canceled Reason
+ Who Cancelled

CANCELLATION REPORT = { Cancellation Data }

CANCELLATION REPORT PARAMETERS = ( Start Date
+ Finish Date)

CANCELLATION NOTICE = Reason Message
+ Reservation Name
+ Reservation Number
+ Mail Confirmation To

CC EXPIRATION DATE = Month-Year

CC NAME = Name
CC NUMBER = 1{Characters}20
CC TYPE = [AX | VI | MC | DC | 2 char code ]
CHECK IN DATE = Date
CHECK IN TIME = Time-Of-Day
CHECK OUT DATE = Date
CHECK OUT TIME = Time-Of-Day
COMMENT = 1{Character}60
COMPANY ADDRESS = Address
COMPANY PHONE = Phone Number
CONFIRMATION = Confirmation Message
   + Confirmation Data
CONFIRMATION DATA = Confirmation Type
   + Reservation Name
   + Reservation Number
   + Mail Confirmation To
CONFIRMATION MESSAGE = 1{Character}1200
CONFIRMATION TYPE = Integer
CREDIT CARD INFORMATION = CC Name
   + CC Type
   + CC Number
   + CC Expiration Date
CURRENT GUEST STATUS = Reservation Number
   + Unit Number
   + Reservation Name
   + Check-In Date
   + Departure Date
   + Number In Party
   + (Group Name)
   + (Number Of Cribs/Hi-Chairs/Roll-a-ways Needed)
   + Nightly Rate
DATE = INTEGER * Produced from Month, Day and Year *
DATE DEPOSIT RECEIVED = Date
DATE STATUS CHANGE = [(Start Date) + (Finish Date)]
DEPARTURE DATA = Reservation Name
+ Unit Number

DEPARTURE DATE = Date

DEPARTURE TIME = Time-Of-Day

DEPOSIT CONFIRMATION = Deposit Confirmation Message
+ Reservation Name
+ Reservation Number
+ Mail Confirmation To

DEPOSIT CONFIRMATION MESSAGE = Confirmation Message

DEPOSIT PAYMENT DATA = Deposit Received
+ Initials
+ (Credit Card Information)

DEPOSIT RECEIVED = Dollar Amount

DEPOSIT REQUEST = Dollar Amount

DEPOSIT REQUIRED = Dollar Amount

DEPOSIT STATUS REQUEST = Reservation Number
+ (Dollar Amount + Deposit Payment Data)

DOLLAR AMOUNT = 3(Digits)10

FINISH DATE = Date

FIRST NAME = 1(Characters)15

GROUP NAME = 1(Character)36

GUEST ARRIVAL REPORT = { Reservation Number
+ Unit Number
+ Reservation Name
+ Arrival Date
+ Arrival Time
+ Number In Party
+ (Group Name)
+ (Number Of Cribs/HI-Chairs/Roll-a-ways Needed)
+ Nightly Rate}

GUEST STATUS CHANGE =
[Departure Date | Nightly Rate | Check-In Date
+ Check-In time | Check-Out Date
+ Check-Out Time | Unit Number]
GUEST STATUS REQUEST = [Reservation Name | Unit Number] + (Guest Status Change)

GUEST'S ADDRESS = Address

GUEST'S COMPANY = 1{Characters}36

GUEST'S PHONE = Phone Number

HEARD ABOUT US THROUGH = 3 Character Code

INITIALS = 2{Characters}3

LAST NAME = 1{Characters}20

LOCK OFF UNITS = 1{Unit Number}10

MAIL CONFIRMATION TO = Name + Address

METHOD OF PAYMENT = ["CA" | "CC" | "MO" | "CK" | Other 2 Char Code]

MIDDLE INITIAL = Character

MISC EQUIPMENT = Comment

MONTH-YEAR = 4 DIGIT * 2 for month and 2 for year *

NAME = Last Name
    + First Name
    + (Middle Initial)

NUMBER IN PARTY = INTEGER

NUMBER OF BEDROOMS = INTEGER

NUMBER OF BEDS = INTEGER

NUMBER OF CHILDREN = INTEGER

NUMBER OF CRIBS NEEDED = INTEGER

NUMBER OF HIGH-CHAIRS NEEDED = INTEGER

NUMBER OF ROLL-A-WAYS NEEDED = INTEGER

PERSON MAKING RESERVATION = Name

PETS = ["Y" | "N"]

PHONE NUMBER = 20 DIGITS
PHONE OF PERSON MAKING RESERVATION = Phone Number

RATE SCHEDULE CLASS = 2 Digit * Varies With Property *

REASON = [ 'MAINT' | 'HSKP' | 'OTHER' ]

REPORT PARAMETERS = [ Cancellation Report Parameters
  | Reservation Report Parameters
  | Unit Status Parameters
  | Arrival Departure Parameters

REPORT REQUEST = Report Type
  + Report Parameters

REPORT TYPE = [ 'C' | 'R' | 'U' | 'A' ]

RESERVATION NAME = Name

RESERVATION CANCELLATION = Reservation Number
  + "Cancelled"

RESERVATION COMMENTS = 2 Comments

RESERVATION CHANGE = Reservation Number
  + ("Canceled")
  + (Reservation Name)
  + (Arrival Date)
  + (Arrival Time)
  + (Departure Date)
  + (Departure Time)
  + (Number In Party)
  + (Group Name)
  + (Number Of Children)
  + (Pets)
  + (Number Of Cribs Needed)
  + (Number Of Roll-a-ways Needed)
  + (Number Of High-Chairs Needed)
  + (Guest's Address)
  + (Guest's Phone)
  + (Guest's Company)
  + (Company Address)
  + (Company Phone)
  + (Person Making Reservation)
  + (Phone of Person Making Reservation)
  + (Mail Confirmation to..)
  + (Billing Information)
  + (Heard About Us Through...)
  + (Reservation Comments)

RESERVATION DATE = Date
RESERVATION DEPOSIT STATUS = Reservation Number
+ Deposit Required
+ Deposit Payment Data
+ Date

RESERVATION NAME = Name

RESERVATION NUMBER = l{Digit}10 + (Character)

RESERVATION RECORD = (see RESERVATION FILE)

RESERVATION REPORT PARAMETERS = (Start Date + Finish Date)

RESERVATION REQUEST = Reservation Name
+ Arrival Date
+ Arrival Time
+ Departure Date
+ Departure Time
+ Number In Party
+ Group Name
+ (Number Of Children)
+ (Pets)
+ (Number Of Cribs Needed)
+ (Number Of Roll-a-ways Needed)
+ (Number Of High-Chairs Needed)
+ (Misc Equipment)
+ Guest's Address
+ Guest's Phone
+ (Guest's Company)
+ (Company Address)
+ (Company Phone)
+ (Person Making Reservation)
+ (Phone of Person Making Reservation)
+ (Mail Confirmation to..)
+ (Billing Information)
+ (Heard About Us Through...)
+ (Reservation Comments)

RESERVATION STATUS = Reservation Number
+ Reservation Name
+ (Group Name)
+ Arrival Date
+ Departure Date
+ Deposit Received

RESERVATIONIST'S INITIALS = 3 Characters

RESERVATIONS IN THIS UNIT = Reservation Number
+ Arrival Date
+ Departure Date

START DATE = Date
TIME-OF-DAY = 4 Digits

UNIT ADDRESS = Address

UNIT DATA = (Unit Number)
+ (Unit Address)
+ (Unit Phone Number)
+ (Number of Bedrooms)
+ (Number Of Beds)
+ (Additional Features)
+ (Rate Schedule Class)
+ (Lock Off Units)

UNIT HOLD = Start Date
+ Finish Date

UNIT NUMBER = 1{Character}10

UNIT PHONE NUMBER = Phone Number

UNIT RELEASE = 'RELEASED'

UNIT SELECTION = Unit Number

UNIT STATUS = [ Usage Report ]

UNIT STATUS CHANGE =
[ Unit Hold + Reason | Unit Release ]
| [ (Start Date) + (Finish Date) ]

UNIT STATUS PARAMETERS = ( Start Date )
+ ( Unit Type )
+ ( {Unit Number} )

UNIT TYPE = [ 1 | 2 | 3 | ... ]

USAGE REPORT = { Unit Number
+ Dates Reserved }
Appendix F

File Descriptions

for the

New Reservations System
CANCELED FILE = {CANCELED RECORDS}

CANCELED RECORD =
  Reservation Number +
  Reservation Name +
  Date of Reservation +
  Reservationist's Initials +
  Unit Number +
  Arrival Date +
  Arrival Time +
  Departure Date +
  Departure Time +
  Number In Party +
  Group Name +
  (Number of Children) +
  (Pets (y/n)) +
  (Number of Cribs Needed) +
  (Number of High-Chairs Needed) +
  (Number of Roll-a-ways Needed) +
  Guest's Address +
  Guest's Phone +
  (Guest's Company) +
  (Company Phone) +
  Nightly Rate +
  Deposit Required +
  Deposit Received +
  Date Deposit Received +
  Method of Deposit Payment +
  (Credit Card Information) +
  (Person Making Reservation) +
  (Phone of Person Making Reservation) +
  (Mail Confirmation to...) +
  (Billing Information) +
  (Heard About us Through...) +
  {{(Comments)}}
  Canceled Date +
  Canceled Reason +
  Canceled Comment +
  Who Called and Canceled +
  Who Took Cancellation
GUEST FILE = {GUEST RECORDS}
GUEST RECORDS =
Reservation Number +
Reservation Name +
Date of Reservation +
Reservationist's Initials +
Unit Number +
Arrival Date +
Arrival Time +
Departure Date +
Departure Time +
Number In Party +
Group Name +
(Number of Children) +
(Pets (y/n)) +
(Number of Cribs Needed) +
(Number of High-Chairs Needed) +
(Number of Roll-a-ways Needed) +
Guest's Address +
Guest's Phone +
(Guest's Company) +
(Company Phone) +
Nightly Rate +
Deposit Required +
Deposit Received +
Date Deposit Received +
Method of Deposit Payment +
(Credit Card Information) +
(Person Making Reservation) +
(Phone of Person Making Reservation) +
(Mail Confirmation to...) +
(Billing Information) +
(Heard About us Through...) +
{(Comments)}
Check-In Date +
Check-In Time
INACTIVE FILE = {INACTIVE RECORDS}
INACTIVE RECORD =
  Reservation Number +
  Reservation Name +
  Date of Reservation +
  Reservationist's Initials +
  Unit Number +
  Arrival Date +
  Arrival Time +
  Departure Date +
  Departure Time +
  Number In Party +
  Group Name +
  (Number of Children) +
  (Pets (y/n)) +
  (Number of Cribs Needed) +
  (Number of High-Chairs Needed) +
  (Number of Roll-a-ways Needed) +
  Guest's Address +
  Guest's Phone +
  (Guest's Company) +
  (Company Phone) +
  Nightly Rate +
  Deposit Required +
  Deposit Received +
  Date Deposit Received +
  Method of Deposit Payment +
  (Credit Card Information) +
  (Person Making Reservation) +
  (Phone of Person Making Reservation) +
  (Mail Confirmation to...) +
  (Billing Information) +
  (Heard About us Through...) +
  ((Comments))
  Check-In Date +
  Check-In Time
  Check-Out Date +
  Check-Out Time
MESSAGE FILE = { MESSAGE RECORD }
MESSAGE RECORD = Message Number
   + Message

PROPERTY FILE = Property Name
   + Property Address
   +{ Rate Schedule Class + Nightly Rate + Weekly Rate}
   +{ Unit Type + Deposit Required + Rate Schedule Class}
RESERVATIONS FILE = {RESERVATIONS RECORDS}
RESERVATIONS RECORD =
    Reservation Number +
    Reservation Name +
    Date of Reservation +
    Reservationist's Initials +
    Unit Number +
    Arrival Date +
    Arrival Time +
    Departure Date +
    Departure Time +
    Number In Party +
    Group Name +
    (Number of Children) +
    (Pets (y/n)) +
    (Number of Cribs Needed) +
    (Number of High-Chairs Needed) +
    (Number of Roll-a-ways Needed) +
    Guest's Address +
    Guest's Phone +
    (Guest's Company) +
    (Company Phone) +
    Nightly Rate +
    Deposit Required +
    Deposit Received +
    Date Deposit Received +
    Method of Deposit Payment +
    (Credit Card Information) +
    (Travel Agent) +
    (Person Making Reservation) +
    (Phone of Person Making Reservation) +
    (Mail Confirmation to...) +
    (Billing Information) +
    (Heard About us Through...) +
    {((Comments))}
UNIT FILE = {UNIT RECORDS}
UNIT RECORD =
    Unit Number +
    Unit Address +
    Unit Phone Number +
    Number of Bedrooms +
    Number of Beds +
    Number of Baths +
    {Additional Features} +
    Rate Schedule Class +
    Unit Description/Comments +
    {Lock Off Units}
    {Reservations in this Unit}

WAITING LIST FILE = { WAITING LIST RECORD }
WAITING LIST RECORD = Reservation Request
Appendix G

Process Descriptions

for

The New System
Accept New Reservation - Process 1

Accept Reservation Request
Process 1.1

For each Reservation Request

1. Accept Reservation Request Data from Customer

2. IF no acceptable units are available
   THEN IF customer desires
       THEN put in waiting list file

Select Unit
Process 1.2

Using the Unit Requirements

1. Find the Least-used unit satisfying the requirements

2. IF None are available
   THEN Display Error Message

Enter Reservation
Process 1.3

Using the Complete Reservation data

1. Tell customer deposit requirements

2. IF customer desires to use Credit Card
   THEN collect Credit Card Information

3. Store Complete Reservation data in Reservation File

4. Update Unit File with Reservation Dates

5. Collect Confirmation Data
Determine Deposit  
Process 1.4

Using the Unit Type  
Access the property file to get the corresponding Deposit Requirements

Generate Confirmation  
Process 1.5

1. Using Confirmation Type  
Retrieve Confirmation Message from Message File

2. Using Confirmation Message and Confirmation Data  
Print out Confirmation
Modify Reservation - Process 2

Get Current Reservation
  Process 2.1

For each Reservation Change

1. Using the Reservation Number from Reservation Change
   IF not a valid Reservation Number
      THEN Display Error Message

2. Retrieve the current reservation data from the Reservation File

Display Current Reservation
  Process 2.2

1. Display Reservation Data Screen

2. Fill in Screen with Current Reservation Data for selected Reservation Number

Alter Reservation Record
  Process 2.3

Using Reservation Change Data

1. IF a cancellation
   THEN Remove Reservation from Reservation File
      Put Reservation into Canceled File
      Update Unit File
   ELSE Make changes in Current Reservation Record
      Make Changes in Unit File

2. Collect Confirmation Data
Generate Confirmation
Process 2.4

1. Using Confirmation Type
   Retrieve Confirmation Message from Message File

2. Using Confirmation Message and Confirmation Data
   Print out Confirmation
Maintain Deposit Status - Process 3

Get Deposit Status
Process 3.1

Using Reservation Number or Name from Deposit Status Request
Get Current Reservation Record from Reservation File

Display Deposit Status
Process 3.2

Using the current Reservation Record
Display the current Deposit status for the Selected Reservation

Update Status
Process 3.3

Using Deposit Payment Data and Reservation Record

1. IF Deposit Received
   THEN Change Deposit Received
       Change Date Deposit Received
       Change Method of Deposit Payment
       IF Credit Card Used
       THEN Change Credit Card Information

2. Save Updated Reservation Record into Reservations File
Generate Confirmation
Process 3.4

1. Using Confirmation Type
   Retrieve Confirmation Message from Message File

2. Using Confirmation Message and Confirmation Data
   Print out Confirmation

Cancel Reservation
Process 3.5

For Each Reservation Cancellation
   IF no Cancellation Approval
   THEN Display Error Message
   ELSE
      Move Reservation Record Data to Canceled File
      Remove Reservation Dates from Unit Record
      Group Cancellation Data together

Generate Cancellation Notice
Process 3.6

1. Retrieve Cancellation Message from Message File

2. Using Cancellation Message and Cancellation Data
   Print out Cancellation Notice
Generate Management Reports - Process 4

=================================================================================================

Determine Type of Report
Process 4.1
=================================================================================================

For all Report Requests
1. Get Report Type

2. IF Report Type is Usage
   THEN Get Unit Report Parameters

3. IF Report Type is Reservation
   THEN Get Reservation Report Parameters

4. IF Report Type is Cancellation
   THEN Get Cancellation Report Parameters

=================================================================================================

Produce Usage Report
Process 4.2
=================================================================================================

Using Unit Status Parameters

For each Unit (lowest usage first)
1. List Unit Information
2. List Unit Status
2. List Unit Usage

=================================================================================================

Produce Reservation Status Report
Process 4.3
=================================================================================================

Using Reservation Report Parameters in selected order

For all Reservations within Parameters
1. List Reservation Information
2. List Effective Dates
3. List Deposit Information
Produce Cancellation Report
Process 4.4

Using Cancellation Report Parameters

For all Reservations within Parameters
1. List Reservation Information
2. List Effective Dates
3. List Deposit Information
4. List Cancellation Information

Produce Arrival-Departure Report
Process 4.5

Using Reservation File

For all Reservations due to arrive This Day
1. List Reservation Information
2. List Effective Dates
3. List Deposit Information

Using Guest File

For all Guests Due to Depart This Day
1. List Guest Information
2. List Room Number
File Maintenance - Process 5

Add Arrivals To Guest File
Process 5.1

Using Reservation File
Move all Reservation data to The Guest File
In Preparation for Check-In

Check for Existing Units
Process 5.2

Using Unit Number from Unit Data
Check Unit File for Existing Unit

Add New Unit
Process 5.3

Using The New Units Data supplied
1. Create a New Unit Record
2. Update the Unit File With New Unit Record

Alter Unit Data
Process 5.4

Using Unit Data supplied
1. Change The selected parts of the Unit Record
2. Update the Unit File with New Unit Record
Purge Files
Process 5.5

Using the File Selected File

1. Read Through The entire File
2. Remove Old Records based on the Purge Request
3. Generate a Hard-Copy of Purged Data (Purge Report)
Maintain Current Guests - Process 6

Get Guest Status
Process 6.1

For Each Guest Status Request
Using Guest Identifier

1. IF invalid Guest Identifier
   THEN Display Error Message

2. Get Guest Record from Guest File

Display Guest Status
Process 6.2

Using the Guest Record
Display the Current Guest Record

Update Unit File
Process 6.3

Using Unit Status Change

1. Get Current Unit Record

2. IF Unit Change does not conflict with another Reservation
   THEN Alter Unit Record and Update File
   ELSE Display Error Message
Change Dates of Stay
Process 6.4

Using Date Status Change

1. Get Current Unit Record

2. IF Date Change does not conflict with another Reservation
   THEN Alter Unit Record and Update File
   Alter Reservation Record and Update File
   ELSE Display Error Message

Alter Guest Status (check out)
Process 6.5

1. Remove Guest from Guest File

2. Save Guest Data in Inactive File

3. Remove Reservation From Reservation File
Appendix H
Structure Charts
for
The New System
Structure Chart
Enter New Reservation
Appendix I

Module Design

for

The New System
Display Start message

DO WHILE Start True
  clear the screen
  Display the Main Menu (see attached)
  DO WHILE choice is not valid (valid choices 0..6)
    Get a choice
  ENDDO

  DO CASE
    CASE choice= 0
      Clear the screen
      Terminate Program
    CASE choice= 1
      Enter a New Reservation
    CASE choice= 2
      Modify a Reservation
    CASE choice= 3
      Deposit Maintenance
    CASE choice= 4
      Management Reports
    CASE choice= 5
      File Maintenance
    CASE choice= 6
      Guest File Maintenance
    OTHERWISE
      DO Print Error Message "Invalid Choice"
  ENDCASE
ENDDO While Start True
RETURN
DO select a unit RETURNS unit selection, unit type (UNITGET)

IF unit selection = 0 (invalid unit number)
    IF customer does not desire to be put on waiting list
        RETURN
    ENDIF
ENDIF

DO accept reservation RETURNS reservation data (RACCEPT)

IF unit selection <> 0
    DO get deposit WITH unit type, RETURNS deposit amt
    DO store reservation WITH reservation data
    DO generate confirmation WITH confirmation data
ELSE
    DO put on waitlist WITH reservation data (WAITLIST)
ENDIF

RETURN
* Module...: Accept Reservation Request (RACCEPT)
* Author....: Warren M. Bartlett
* Date......: 07/15/84
* Notice....: Copyright 1984, All Rights Reserved
* Notes.....: Gets values in the Reservation Screen
* Calls.....: RESVSCRN to display screen
* Parameters Reservation Data
*
Clear values in all temporary variables

DO display Reservation Screen

DO WHILE NOT valid reservation information

Get all information for reservation

IF NOT valid information
    DO PRINT Error Message
ENDIF

ENDDO

RETURN
* Module...: Generate A Confirmation Letter GENCONF
* Author...: Warren M. Bartlett
* Date.....: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Calls.....: NONE
* Called by: Various

* Parameters Message Type (A | B | ...),
  Confirmation Data

Using Message File Find Appropriate Message
Print Confirmation Data
Print Confirmation Message
Eject Page from printer
RETURN
Using Unit Data File (UDATA)

Find unit type

RETURN deposit amount
DO display unit screen

USING unit data file (UDATA)

DO WHILE NOT EOF()
    skip a line
    Display Unit Descriptions
ENDDO

DO WHILE NOT got a unit

    GET unit type

    DO find a unit WITH unit selection, unit type

    IF No unit of this type available
        RETURN unit selection = 0
    ELSE
        RETURN unit type and unit number
    ENDIF

ENDDO

RETURN
Find Least used unit of type "unit type"

IF NOT Unit found

    RETURN Unit selection Zero (indicates no unit found)

ELSE

    RETURN Unit Number Selected

ENDIF

RETURN
* Module....: Print Error Message - ERRORM
* Author....: Warren M. Bartlett
* Date......: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Calls....: NONE
* Called by: Various
*
* Parameters Message to display
*
Ring Bell
Display Message
Delay for Reading
Clear Message Area
RETURN
DO Generate Reservation Number RETURNS Reservation Number

Using the reservation file
Append a new record
Replace fields with complete reservation
Using Unit Reservations file
Append new field

Fill in start and stop dates and Reservation number
Using Unit file add days reserved to total

RETURN
* Module...: Generate a New Reservation Number (GRESNUM)
* Author...: Warren M. Bartlett
* Date.....: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls.....: NONE
* Called by: STORRES
* Parameters Reservation Number
*
FROM Reservation Number File

Recall Last Used Reservation Number

IF Reservation Number = Maximum
   Reservation Number = 1
ELSE
   Increment Reservation Number
ENDIF

Save New Reservation Number

RETURN New Reservation Number
* Module...: Put reservation in Waiting List file (WAITLIST)
* Author...: Warren M. Bartlett
* Date.....: 07/17/84
* Notice...: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls....: NONE
* Called by: RESENTER
* Parameters Reservation data

Using the waiting list file

Append a new record

Replace fields with reservation data

RETURN
* Module...: Modify Existing Reservation (MODIFRES)
* Author....: Warren M. Bartlett
* Date......: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls.....: GETRESKY to get the Reservation Key
*           : GETCURES to get current reservation
*           : DISPRES to display the Reservation
*           : CANRES to cancel the Reservation
*           : CHANGRES to change the Reservation
*           : GENCONF to generate a new confirmation
* Called by: MENU
* * Parameters None
*

DO Get Reservation Key (RETURNS Number or Name)

IF NO KEY

    RETURN
ELSE

    DO Get Current Reservation WITH Reservation Key
       RETURNS Reservation Record

    IF No reservation

        RETURN
ELSE

    DO Display Reservation WITH Reservation Record
   Prompt user for Cancellation or Change
    IF user does NOT want a cancellation
       DO Change Reservation WITH Reservation Record
    ELSE
       DO Cancel Reservation WITH Reservation Record
       ENDIF
    DO Generate Confirmation WITH
       Message Type and Confirmation Data
       ENDIF
ENDIF

RETURN
* Module...: Get current reservation (GETCURES)
* Author....: Warren M. Bartlett
* Date......: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls....: None
* Called by: MENU
* Parameters Reservation Key, Reservation Record
*
  Using Reservation File

  Find Reservation Record With Reservation Key

  IF NOT found

    DO Print Error Message With Row, Column, and Message
    RETURN No Reservation Record

  ELSE

    RETURN Reservation Record

ENDIF
Clear Screen

Prompt User for Reservation Number

IF NO Number

    Prompt for Reservation Name
    RETURN Name

ELSE

    RETURN Number

ENDIF
* Module....: Display Current Reservation (DISPRES)
* Author....: Warren M. Bartlett
* Date......: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls.....: RSCREEN to display Reservation Screen
* Called by: MODIFRES
* Parameters Reservation Record
*
DO Display Reservation Screen

Display Reservation Data for Screen

RETURN
* Module....: Change Reservation Data (CHGRSDAT)
* Author....: Warren M. Bartlett
* Date......: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls.....: NONE
* Called by: MODIFRES
* Parameters Reservation Record
*
Display Edit Instructions
Allow Editing of Reservation Data
Replace reservation record in Reservation File
IF Dates of Stay are Altered
    Replace Unit Reservation Dates
    Adjust Unit’s Days Reserved
ENDIF
RETURN
DO Get Cancellation Information WITH Cancellation Info
Remove Unit Dates from Unit reservation file
Subtract the number of Days from Unit's Days Reserved
Mark Reservation as Deleted
DO PUT Reservation Record in Canceled file
DO Check Waiting List
RETURN
Get additional Information for file from user.
1. Date Canceled
2. Reason Canceled
3. Comment (if necessary)
4. Who Called and Canceled
5. Who Took Cancellation

RETURN
Append a record to Canceled file

Copy Reservation file fields to Canceled Record

Store Cancellation Info in Canceled Record

RETURN
DO Get Reservation Key (RETURNS Number or Name)

IF NO KEY
    RETURN
ELSE

    DO Get Current Reservation WITH Reservation Key
        RETURNS Reservation Record

    IF No reservation
        RETURN
    ELSE

        DO Display Deposit Status WITH Deposit Data
        Prompt user for Cancellation or Change
        IF user does NOT want a cancellation
            DO Change Deposit Status
        ELSE
            DO Cancel Reservation WITH Reservation Record
        ENDIF
        DO Generate Confirmation WITH
            Message Type and Confirmation Data

    ENDF
ENDIF

RETURN
DO Display Deposit Screen

Display Deposit Data on Screen

RETURN
Display Edit Instructions

Allow Editing of Deposit Data

Replace Reservation record in Reservation File

RETURN
DO WHILE True
    clear the screen
    Display the Report Menu (see attached)
    DO WHILE choice is not valid (valid choices 0..4)
        Get a choice
    ENDDO

    DO CASE
        CASE choice= 0
            Clear the screen
            RETURN
        CASE choice= 1
            DO Cancellation Report
        CASE choice= 2
            DO Arrival Departure Report
        CASE choice= 3
            DO Reservation Status Report
        CASE choice= 4
            DO Usage Report
        OTHERWISE
            DO Print Error Message "Invalid Choice"
    ENDCASE

ENDDO While True

RETURN
* Module...: Do Cancellation Report (CANCEPT)
* Author...: Warren M. Bartlett
* Date.....: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved..: NONE
* Calls.....: NONE
* Called by: REPRTGEN
* Parameters: None
*

Prompt User for Start and End Date

Using Cancellation File in Date Order

Print Heading at top of page
1. Report Name
2. Start and End Date
3. Date of Report
4. Page Number
5. Column Labels

DO WHILE NOT EOF

IF Cancellation Date between Start and End Date

Print
1. Reservation Number
2. Reservation Name
3. Date Canceled
4. Reason Canceled
5. Who Took Cancellation

ENDIF

SKIP Record

ENDDO

EJECT Last Page

RETURN
* Module...: Print Arrival/Departure Report
* Author...: Warren M. Bartlett
* Date.....: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls.....: NONE
* Called by: REPRGEN
* Parameters None
*

Prompt User for the Date for the Report
Using Reservation
Print Heading at top of page
   Rept Name, Date of Arrival, Rept Date, Pg Number,
   and Column Labels

DO WHILE NOT EOF
   IF Arrival Date equals Date for Report
   Print
       1. Reservation Number
       2. Reservation Name
       3. Unit Number
       4. Time Due
       5. Number in Party
       6. Cribs, High-Chairs, Roll-a-ways
       7. Comments
   ENDIF
   SKIP Record
ENDDO
EJECT Last Page

Using Guest File
Print Heading at top of page
   Rept Name, Date of Departure, Rept Date, Pg Number,
   and Column Labels

DO WHILE NOT EOF
   IF Departure Date equals Date for Report
   Print
       1. Reservation Number
       2. Reservation Name
       3. Unit Number
       4. Time Due Out
       5. Cribs, High-Chairs, Roll-a-ways
       7. Comments
   ENDIF
   SKIP Record
ENDDO

EJECT Last Page
RETURN
Prompt User for Start and End Date

Using Reservation File in Date Order

Print Heading at top of page
1. Report Name
2. Start and End Date
3. Date of Report
4. Page Number
5. Column Labels

DO WHILE NOT EOF

IF Arrival Date between Start and End Date

Print
1. Reservation Number
2. Reservation Name
3. Arrival Date
4. Deposit Required
5. Deposit Received

ENDIF

SKIP Record

ENDDO

EJECT Last Page

RETURN
* Module....: Usage Report (USEREPT)
* Author....: Warren M. Bartlett
* Date......: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Notes.....:
* Reserved.: NONE
* Calls.....: NONE
* Called by: REPRGEN
* Parameters None
*
Sort Unit File into Order by:
  1. Unit Type
  2. Inverse Order by Days Reserved

Using Unit File

WHILE NOT EOF
  For Each Type of Unit:

    Print Heading at top of page
    1. Report Name
    2. Unit Type
    3. Start and End Date
    4. Date of Report
    5. Page Number
    6. Column Labels

    Print out Each Unit Number and Days Reserved.

    Eject Page

ENDDO

RETURN
DO WHILE True
   clear the screen
   Display the File Maintenance Menu (see attached)
   DO WHILE choice is not valid (valid choices 0..4)
      Get a choice
   ENDDO
   DO CASE
      CASE choice = 0
         Clear the screen
         RETURN
      CASE choice = 1
         DO Backup Data Files
      CASE choice = 2
         DO Add Arrivals to Guest File
      CASE choice = 3
         DO Maintain Units
      CASE choice = 4
         DO Purge Files
      OTHERWISE
         DO Print Error Message "Invalid Choice"
   ENDCASE
   ENDDO While True
RETURN
* Module...: Backup Files (BACKUP)
* Author...: Warren M. Bartlett
* Date.....: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved..: NONE
* Calls....: NONE
* Called by: FILMAINT
* Parameters None
*

Provide Necessary Instructions for User to Backup all Data files

Do the necessary Commands

RETURN
* Module...: Add Scheduled Arrivals to Guest File (ADDARRVL)
* Author...: Warren M. Bartlett
* Date.....: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls.....: NONE
* Called by: FILMAINT
* Parameters None
*
Using Reservation File

DO WHILE NOT EOF
    IF Arrival Date Equals Today
        Copy Record from Reservation File to Guest File
    ENDF
    SKIP Record
ENDDO
RETURN
Prompt User To ADD a Unit or EDIT a Unit

DO Display Unit Screen (See Appendix J.)

IF ADD

Get and Display Unit Information
ADD new Record to Unit File

ELSE

Prompt For Unit Number
Using Unit File
DO Display Current Information
Allow Editing of Information
REPLACE Record in Unit File

ENDIF

RETURN
* Module...: Purge Files
* Author...: Warren M. Bartlett
* Date.....: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.:. NONE
* Calls....: NONE
* Called by: FILMAINT
* Parameters None
*
Select Date for Records older than that to be deleted.

Select File to Purge through Menu

Ask If The User is Absolutely Sure!

Purge Selected File Printing Hard copy of Purged Records

RETURN
Get Guest Status Request (Name, Unit, or Res Number)

With Guest File Find Guest Record

IF NO Record
    Print Error Message
    RETURN
ENDIF

DO Display Guest Status Screen (See Appendix J.)

DO Display Guest Current Status

Prompt For
    1. Unit Change
    2. Date Change
    3. Check In/Out

DO WHILE choice is not valid (valid choices 0..3)
    Get a choice
ENDDO

DO CASE
    CASE choice= 0
        Clear the screen
        RETURN
    CASE choice= 1
        DO Unit Change (GSTUCHG)
    CASE choice= 2
        DO Date Change (GSTDCHG)
    CASE choice= 3
        DO Check In/Out (GSTINOUT)
ENDCASE

RETURN
* Module....: Unit Change (GSTUCHG)
* Author....: Warren M. Bartlett
* Date......: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls....: NONE
* Called by: FILGUEST
* Parameters None
*
Get Unit Number from User

Get Dates If different from Current Departure Date

Using Unit File

IF New Unit is NOT available for Dates Requested

    Print Error Message
    RETURN
ENDIF

Change Unit Records for Current Unit and New Unit

Change Guest File Record

Change Reservation File Record

RETURN
Get New Departure Date From User

IF New Date Later than Current Date

    Check Unit File to Determine if Unit is Free
    IF Unit NOT Free

        Print Error Message
        RETURN

ENDIF
ENDIF

Change Unit Record for Current Unit
Change Guest File Record
Change Reservation File Record

RETURN
Prompt for Check In or Check Out

IF Check In

    Put Date and Time into Guest Record
    Update Guest Record

ELSE

    Append Record to Inactive File
    Put Guest Record in Inactive Record
    Put Check out Date and Time in Inactive Record
    Delete Guest Record

ENDIF

RETURN
Appendix J

Screen Design

for

The New System
Main Menu

0. EXIT
1. Enter a New Reservation
2. Modify a Reservation
3. Deposit Maintenance
4. Management Reports
5. File Maintenance
6. Guest Maintenance

Enter Your Choice:
The Reservation System
Copyright (c) 1984
Warren M. Bartlett

Report Menu

0. EXIT
1. Cancellation Report
2. Arrival/Departure Report
3. Reservation Status Report
4. Usage Report

Enter Your Choice:

File Maintenance Menu

0. EXIT
1. Backup Data Files
2. Add Arrivals to Guest File
3. Maintain Unit Data File
4. Purge Files

Enter Your Choice:

Screen Design
Other Menus
### The Reservation System

**Screen 1**

**MODE: Enter a New Reservation**

<table>
<thead>
<tr>
<th>Last Name:</th>
<th>First Name:</th>
<th>MI:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Name:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Company:</th>
<th>Company Phone:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Co. Address:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>City:</td>
<td>State:</td>
<td>Zip:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Home Address:</th>
<th>Home Phone:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>City:</td>
<td>State:</td>
<td>Zip:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arrival:</th>
<th>Time:</th>
<th>Departure:</th>
<th>Time:</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Size of Party:</th>
<th>Children:</th>
<th>Pets (Y or N):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Cribs:</td>
<td>Roll-a-ways:</td>
<td>High Chairs:</td>
</tr>
<tr>
<td>Misc. Equipment Needed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UNIT NUMBER</th>
<th>NIGHTLY RATE</th>
<th>DEPOSIT REQUIRED</th>
</tr>
</thead>
</table>
The Reservation System

Screen 2

MODE: Enter a New Reservation

Bill To: Name:
Address:
City: State: Zip:

Mail Confirmation To: Name:
Address:
City: State: Zip:

Person Making Res: Phone #
Travel Agent Name:

Deposit Information: Deposit Required
Deposit Received: How Paid:
The Reservation System

MODE: Unit Selection

Unit Types Available:
A. 
B. 
C. 
D. 
E. 
F. 
G. 
H. 
I. 
J. 
K. 
L. 
M. 
N. 
O. 
Enter Unit Type Desired (RETURN to Quit):
The Reservation System

MODE: Maintain Deposit

Reservation Number:
Reservation Name:
Home Address:
City: State: Zip Code:

Company Name:
Address:
City: State: Zip Code:

Deposit Required:
Deposit Received: Date Received:
How Paid:

Credit Card Information
CC Type:
CC Number: Expires:
Appendix K

Data Base Files

for

The New System
Structure for database : C:CANCELLD.dbf

<table>
<thead>
<tr>
<th>Field</th>
<th>Field name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RES_NUMBER</td>
<td>Character</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>RES_DATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>WHO_TOOK</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>RES_UNIT</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>LNAME</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>FNAME</td>
<td>Character</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MI</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>GNAME</td>
<td>Character</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>COMPANY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>CADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>PHONE</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>CITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>STATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>ZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>HADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>PHONE</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>CITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>STATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>ZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>ADATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>ATIME</td>
<td>Character</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>DDATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>DTIME</td>
<td>Character</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>SIZE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>CHILDREN</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>PETS</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>RIBS</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>ROLLAWAY</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>CHAIR</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>XEQUIP</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>BNAME</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>BADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>BCITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>BSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>ZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>CNAME</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>CNADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>CNCITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>CNSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>CNZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>WHOMADE</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>WHATPHONE</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>NIGHTRATE</td>
<td>Numeric</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>44</td>
<td>DEPST_REQ</td>
<td>Numeric</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>45</td>
<td>DEPST_RECV</td>
<td>Numeric</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>46</td>
<td>DEPST_DATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>DEPST_HOW</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>TRAVEL_AGN</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>-------</td>
<td>-------</td>
<td>-----</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>COMMENTS</td>
<td>Memo</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>CAN_DATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>CAN_REASON</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>CAN_WHO</td>
<td>Character</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>CAN_BY</td>
<td>Character</td>
<td>36</td>
<td></td>
</tr>
</tbody>
</table>

**Total** 809
Structure for database: C: CREDITCD.dbf

<table>
<thead>
<tr>
<th>Field</th>
<th>Field name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RES_NUMBER</td>
<td>Character</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>LNAME</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>FNAME</td>
<td>Character</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MI</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CC_TYPE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>CC_NUMBER</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CC_EXPIR</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

===============================================

** Total **

78
### Structure for database: C:GUEST.dbf

<table>
<thead>
<tr>
<th>Field</th>
<th>Field name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RES_NUMBER</td>
<td>Character</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>RES_DATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>WHO_TOOK</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>RES_UNIT</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>LNAME</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>FNAME</td>
<td>Character</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MI</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>GNAME</td>
<td>Character</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>COMPANY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>CADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>CPHONE</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>CCITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>CSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>CZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>HADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>HPHONE</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>HCITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>HSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>HZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>ADATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>ATIME</td>
<td>Character</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>DDATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>DTIME</td>
<td>Character</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>PSIZE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>CHILDREN</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>PETS</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>CRIBS</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>ROLLAWAY</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>HCHAIR</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>XEQUIP</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>BNAME</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>BADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>BCITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>BSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>BZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>CNNAME</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>CNADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>CN_CITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>CNSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>CNZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>WHOMADE</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>WPHONE</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>NIGHTRATE</td>
<td>Numeric</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>44</td>
<td>DEPST_REQ</td>
<td>Numeric</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>45</td>
<td>DEPST_RECV</td>
<td>Numeric</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>46</td>
<td>DEPST_DATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>DEPST_HOW</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>TRAVEL_AGN</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>COMMENTS</td>
<td>Memo</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>
50  CKIN_DATE  Date   8
51  CKIN_TIME  Character  4

====================================
** Total **                 713
Structure for database: C:INACTIVE.dbf

<table>
<thead>
<tr>
<th>Field</th>
<th>Field name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RES_NUMBER</td>
<td>Character</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>RES_DATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>WHO_TOOK</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>UNIT_NUM</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>LNAME</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>FNAME</td>
<td>Character</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MI</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>GNAME</td>
<td>Character</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>COMPANY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>CADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>CPHONE</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>CCITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>CSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>CZIP</td>
<td>Character</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>HADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>HPHONE</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>HCITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>HSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>HZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>ADATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>ATIME</td>
<td>Character</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>DDATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>DTIME</td>
<td>Character</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>PSIZE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>CHILDREN</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>PETS</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>CRIBS</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>ROLLAWAY</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>HCHAIR</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>XEQUIP</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>BNAME</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>BADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>BCITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>BSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>BZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>CNNAME</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>CNADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>CNCITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>CNSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>CNZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>WHOMADE</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>WHOPHONE</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>NIGHTRATE</td>
<td>Numeric</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>44</td>
<td>DEPST_REQ</td>
<td>Numeric</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>45</td>
<td>DEPST_RECV</td>
<td>Numeric</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>46</td>
<td>DEPST_DATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>DEPST_HOW</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>TRAVEL_AGN</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>COMMENTS</td>
<td>Memo</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>
Structure for database : C:LOCKOFF.dbf

Field | Field name   | Type    | Width | Dec
---|--------------|---------|-------|---
1 | UNIT_NUM     | Character | 10    |   
2 | LKOFF_NUM    | Character | 10    |   

** Total ** 21

Structure for database : C:MESSAGE.dbf

Field | Field name | Type    | Width | Dec
---|------------|---------|-------|---
1 | M_NUMBER   | Character | 2     |   
2 | MESSAGE    | Memo    | 10    |   

** Total ** 13

Structure for database : C:PDATA.dbf

Field | Field name   | Type    | Width | Dec
---|--------------|---------|-------|---
1 | P_NAME_1     | Character | 30    |   
2 | P_NAME_2     | Character | 30    |   
3 | P_NAME_3     | Character | 30    |   
4 | P_ADDRESS1   | Character | 40    |   
5 | P_ADDRESS2   | Character | 40    |   
6 | P_CITY       | Character | 20    |   
7 | P_STATE      | Character | 2     |   
8 | P_ZIP        | Character | 10    |   
9 | P_PHONE      | Character | 20    |   
10 | P_MANAGER    | Character | 35    |   

** Total ** 258
Structure for database: C:RESERVAT.dbf

<table>
<thead>
<tr>
<th>Field</th>
<th>Field name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RES_NUMBER</td>
<td>Character</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>RES_DATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>WHO_TOOK</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>RES_UNIT</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>LNAME</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>FNAME</td>
<td>Character</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MI</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>GNAME</td>
<td>Character</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>COMPANY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>CADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>CPHONE</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>CCITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>CSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>ZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>HADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>PHONE</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Hcity</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>HSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>ZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>ADATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>ATIME</td>
<td>Character</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>DATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>DTIME</td>
<td>Character</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>PSIZE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>CHILDREN</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>PETS</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>CRIBS</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>ROLLAWAY</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>HCHAIR</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>XEQUIP</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>BNAME</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>BADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>BCITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>BSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>BZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>CNNAME</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>CNADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>CNCity</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>CNSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>CNZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>WHOMADE</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>WHOPHONE</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>NIGHTRATE</td>
<td>Numeric</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>44</td>
<td>DEPST_REQ</td>
<td>Numeric</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>45</td>
<td>DEPST_RECV</td>
<td>Numeric</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>46</td>
<td>DEPST_DATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>
** Total ** 701

Structure for database : C:UDATA.dbf

<table>
<thead>
<tr>
<th>Field</th>
<th>Field name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U_TYPE</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>U_DESCRIP</td>
<td>Character</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>U_DEPOSIT</td>
<td>Numeric</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>U_NIGHTRAT</td>
<td>Numeric</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

** Total ** 54

Structure for database : C:UNIT.dbf

<table>
<thead>
<tr>
<th>Field</th>
<th>Field name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U_NUM</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>U_ADDR</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>U_CITY</td>
<td>Character</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>U_STATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>U_ZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>U_PHONE</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>U_BEDROOMS</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>U_BEDS</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>U_BATHS</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>U_TYPE</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>U_DESCRIP</td>
<td>Character</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>U_LOCKOFF</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>U_DAYS</td>
<td>Numeric</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

** Total ** 127

Structure for database : C:UNITRES.dbf

<table>
<thead>
<tr>
<th>Field</th>
<th>Field name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RES_NUMBER</td>
<td>Character</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>U_NUM</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>START_DATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>END_DATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

** Total ** 38
Structure for database: C:WAITLIST.dbf

<table>
<thead>
<tr>
<th>Field</th>
<th>Field name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>WHO_TOOK</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>LNAME</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>FNAME</td>
<td>Character</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MI</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>GNAME</td>
<td>Character</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>COMPANY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>CPHONE</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>CCITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>CSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>CZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>HADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>HPHONE</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>HCITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>HSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>HZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>ADATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>ATIME</td>
<td>Character</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>DDATE</td>
<td>Date</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>DTIME</td>
<td>Character</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>PSIZE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>CHILDREN</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>PETS</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>CRIBS</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>ROLLAWAY</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>HCHAIR</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>XEQUIP</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>BNAME</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>BADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>BCITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>BSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>BZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>CNNAME</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>CNADDRESS</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>CNCITY</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>CNSTATE</td>
<td>Character</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>CNZIP</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>WHOMADE</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>WHOPHONE</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>COMMENTS</td>
<td>Memo</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>TRAVEL_AGN</td>
<td>Character</td>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>

**Total** 642
Appendix L

Implementation of

The New System
* Program...: MENU.PRG
* Author....: Warren M. Bartlett
* Date......: 07/15/84
* Notice....: Copyright 1984, All Rights Reserved
* Notes.....:
* Reserved.: select, selectnum
*
PUBLIC rsvname
PUBLIC Xrnum,Xrdate,Xunitnum,Xlname,Xfname,Xmi,Xgname
PUBLIC Xcompany,Xrsvname,Xdepdate,XCphone,Xcadd,Xccity,
PUBLIC Xcstate,Xczip,Xaddress,Xphone,Xcity,Xstate,Xzip
PUBLIC Xaddate,Xatime,Xddate,Xdtime,Xpsize,Xchildren
PUBLIC Xpets,Xcribs,Xrollaway,Xchair,Xequip,Xbname
PUBLIC Xbadress,Xbcity,Xbstate,Xbzip,Xcnname,Xcnadress
PUBLIC Xcncity,Xcnstate,Xcnzip,Xmade,Xmadephone,Xagent
PUBLIC Xunitdep,Xdeprecvd,Xdephow

SET ECHO OFF
SET TALK OFF
SET BELL OFF

* Note: Debugging Code follows....
*
IF "U" <> TYPE("DEBUG")
   DANSWER = " "
   WAIT "PRINTER? (Y/N) " TO DANSWER
   IF DANSWER = "Y"
      SET DEBUG ON
   ELSE
      SET DEBUG OFF
   ENDIF
   SET TALK ON
   SET ECHO ON
   RELEASE DANSWER
ELSE
   DO COPYRIGHT
ENDIF

* Get Reservationist's Name
DO WHILE .T.
   rsvname = " "
   @ 19,05 SAY "Enter Reservationist's Name: "
   @ 19,35 GET rsvname PICTURE "!XXXXXXXXXXXXXXXXXXXXXXX"
   READ
   IF rsvname <> " "
      EXIT
   ENDIF
   ? chr(7)
   @ 20,35 SAY "Please Enter Your Name to Continue!"
ENDDO
* Get Menu Selection
DO WHILE .T.
SET FORMAT TO MAINMENU
STORE 7 TO selectnum
DO WHILE selectnum < 0 .OR. selectnum > 6
STORE "0" TO select
READ
STORE VAL(select) TO selectnum
ENDDO
SET FORMAT TO
DO CASE
CASE selectnum= 0
SET BELL ON
SET TALK ON
CLEAR ALL
CLEAR
RETURN
CASE selectnum= 1
* DO 1. Enter a New Reservation
DO RESENTER
CASE selectnum= 2
* DO 2. Modify a Reservation
DO MODIFRES
CASE selectnum= 3
* DO 3. Deposit Maintenance
DO DEPMAINT
CASE selectnum= 4
* DO 4. Management Reports
DO REPRTGEN
CASE selectnum= 5
* DO 5. File Maintenance
DO FILMAINT
CASE selectnum= 6
* DO 6. Guest File Maintenance
DO FILGUEST
ENDCASE
ENDDO
* EOF: Menu.PRG
set color to W+/B+,R+/ +, +

CLEAR

@ 1,17 SAY "|============================================================|
@ 2,17 SAY "| |============================================================|
|"
@ 3,17 SAY "| |
| |
@ 4,17 SAY "| |
| |
@ 5,17 SAY "| |
| |
@ 6,17 SAY "| |
| |
@ 7,17 SAY "| |
| |
@ 8,17 SAY "| |
| |
@ 9,17 SAY "| |
| |
@ 10,17 SAY "| |
| |
@ 11,17 SAY "| |
| |
@ 12,17 SAY "| |============================================================|
|"
@ 13,17 SAY "|============================================================|

@ 4,27 SAY "The Condo Reservation System"
@ 6,22 SAY "Copyright (c) Warren M. Bartlett 1984"
@ 8,26 SAY "As a Licensed Proprietary Work"
@ 10,30 SAY "All Rights Reserved"

RETURN
Program...: RESENTER.PRG - Enter a Reservation
Author....: Warren M. Bartlett
Date......: 07/17/84
Notice....: Copyright 1984, All Rights Reserved
Notes.....: Main Driver for Accepting new Reservations
Reserved.: NONE
Calls.....: unitget, raccept, rgetdep, rstorres, genconf
Called by: menu
Local.....: unitnum, unittype, deposamt

PUBLIC unitnum, unittype, unitdep, unitrate, unitdesc,
  errflag
unitnum = "0"
unitdesc = " "
unittype = " "
unitdep = 0
unitrate = 0
Xadate = " / / "
Xddate = " / / "
errflag = .F.
DO unitget WITH unitnum, unitdesc, unittype, Xadate, Xddate
  IF unitnum <> "0"
    DO rgetdep WITH unittype, unitdep, unitrate
    DO raccept WITH errflag
    IF errflag
      RETURN
    ENDIF
  ENDIF
  DO rstorres
  DO genconf
ENDIF
ELSE
  IF unittype = "W"
    DO waitlist
  ENDIF
ENDIF
RELEASE ALL EXCEPT X*
RETURN
PARAMETERS errflag

DO clearvar

DO rsvscrnl

DO dispunit

DO rgetdat1

DO rsvscrn2

DO dispunit

DO rgetdat2

RETURN
clear
@ 1,0 SAY "|================================|"
    "="
@ 1,56 SAY "|================================|
    "="
@ 2,0 SAY "| The Reservation System |
| MODE: "
@ 2,55 SAY "|Unit Selection |
    "|
@ 3,0 SAY "|================================|
    "="
@ 3,56 SAY "|================================|
    "="
@ 4,0 SAY "|================================|
    "="
@ 4,56 SAY "|================================|
    "="
@ 5,0 SAY "| Unit Types Available:"
@ 5,49 SAY "|Deposit"
@ 5,65 SAY "|Nightly Rate"
@ 5,78 SAY "|
@ 6,0 SAY "|================================|
    "="
@ 6,56 SAY "|================================|
    "="
@ 7,0 SAY "|
@ 7,78 SAY "|
@ 8,0 SAY "|
@ 8,78 SAY "|
@ 9,0 SAY "|
@ 9,78 SAY "|
@ 10,0 SAY "|
@ 10,78 SAY "|
@ 11,0 SAY "|
@ 11,78 SAY "|
@ 12,0 SAY "|
@ 12,78 SAY "|
@ 13,0 SAY "|
@ 13,78 SAY "|
@ 14,0 SAY "|
@ 14,78 SAY "|
@ 15,0 SAY "|
@ 15,78 SAY "|
@ 16,0 SAY "|
@ 16,78 SAY "|
@ 17,0 SAY "|
@ 17,78 SAY "|
@ 18,0 SAY "|
@ 18,78 SAY "|
Enter Unit Type Desired (RETURN to Quit):
* Program...: UNITGET.PRG - Get the Unit for the Reservation
* Author...: Warren M. Bartlett
* Date.....: 07/15/84
* Notice....: Copyright 1984, All Rights Reserved
* Notes....:
* Calls....: ugetscrn to display screen
* unitfind to find a unit that satisfies criteria
* getdates to get arrival and departure dates
* Called by: resenter
*
PARAMETERS unitnum, unitdesc, unittype, Xadate, Xddate
PRIVATE maxunit, utype, gotunit
*
* display screen
DO ugetscrn
*
* get unit_type
uline = 7
USE udata
DO WHILE .NOT. EOF()
  @ uline,4 SAY U_TYPE+.U_DESCRIP
  @ uline,50 SAY U_DEPOSIT
  @ uline,68 SAY U_NIGHTRAT
  skip
  uline = uline + 1
ENDDO
skip -1
maxunit = U_TYPE
USE
gotunit = .F.
DO WHILE .NOT. gotunit
  utype = " "
  @ 23,1 SAY "Enter Unit Type Desired (RETURN to Quit)"
  @ 23,51 GET utype PICTURE "!"
  READ
  DO CASE
  CASE utype = " "
    unitnum = "0"
    gotunit = .T.
  CASE utype < 'A'
    ?? chr(7)
    utype = " "
    @ 23,1 SAY "  Invalid Choice... Press Enter Key
to continue" GET utype
    READ
  CASE utype > maxunit
    ?? chr(7)
    utype = " "
    @ 23,1 SAY "  Invalid Choice... Press Enter Key
to continue" GET utype
READ
OTHERWISE
   DO getdates WITH Xadate, Xddate
   DO unitfind WITH unitnum, unitdesc, utype, Xadate, Xddate
   IF unitnum = "0"
      wanswer = "N"
      @ 23,1 SAY "Unit type Not available Put on Wait List (Y or N)? "
      @ 23,59 GET wanswer PICTURE "!"
      read
      IF wanswer = "Y"
         gotunit = .T.
         unittype = "W"
      ENDIF
   ELSE
      @ 23,1 SAY "Unit Selected : "
      @ 23,50 SAY unitnum
      unittype = utype
      gotunit = .T.
   ENDIF
ENDCASE
ENDDO
RETURN
* Program...: Generate Reservation Number (GRESNUM)
* Author....: Warren M. Bartlett
* Date.....: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Notes.....:
* Reserved.: NONE
* Calls....: NONE
* Called by: STORRES
*
PARAMETERS resnumber

RESTORE FROM rsnum ADDITIVE

IF lastnumber >= 999999
    lastnumber = 100000
ELSE
    lastnumber = lastnumber + 1
ENDIF

resnumber = STR(lastnumber,6)

SET SAFETY OFF

SAVE TO rsnum ALL LIKE lastnumber

SET SAFETY ON

RELEASE lastnumber

RETURN
* Program...: Put Reservation into Wait List (WAITLIST)
* Author....: Warren M. Bartlett
* Date......: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Notes.....:
* Reserved.: NONE
* Calls.....: NONE
* Called by: RESENTER
*
USE waitlist INDEX waitname
APPEND BLANK

REPLACE LNAME WITH Xlname
REPLACE FName WITH Xfname, MI WITH Xmi
REPLACE CNAME WITH Xgname, COMPANY WITH Xcompany
REPLACE CPHONE WITH XCphone, CADDRESS WITH Xcadd
REPLACE CCITY WITH Xccity, CSTATE WITH Xcstate
REPLACE CZIP WITH Xczip, HADDRESS WITH Xaddress
REPLACE HPHONE WITH Xphone, HCITY WITH Xcity
REPLACE HSTATE WITH Xstate,
REPLACE ADATE WITH CTOD(Xadate), HZIP WITH Xzip
REPLACE DDATE WITH CTOD(Xddate),
REPLACE PSIZE WITH Xpsize,
REPLACE PETS WITH Xpets,
REPLACE ROLLLAwy WITH Xrollaway,
REPLACE BNAME WITH Xbname,
REPLACE BCITY WITH Xbcity,
REPLACE BZIP WITH Xbzip,
REPLACE CNADDRESS WITH Xcnaddress,
REPLACE CNSTATE WITH Xcncstate,
REPLACE WHOMADE WITH Xmade,
REPLACE TRAVEL AGN WITH Xagent,
REPLACE DEPST_REQ WITH Xunitdep,
REPLACE DEPST_DATE WITH DATE(),

USE

RETURN
rnum = " "

* Generate a Reservation Number
DO gresnum WITH rnum

IF Xdeprecvd <> 0
  DO credcard WITH rnum
ENDIF

USE reservat index resnum, resname
APPEND BLANK
REPLACE RES_NUMBER WITH rnum, RES_DATE WITH DATE()
REPLACE RES_UNIT WITH unitnum, LNAME WITH Xlname
REPLACE FN NAME WITH Xfname, MI WITH Xmi
REPLACE GNAME WITH Xgname, COMPANY WITH Xcompany
REPLACE CPHONE WITH Xphone, CADDRESS WITH Xcadd
REPLACE CCITY WITH Xccity, CSTATE WITH Xcstate
REPLACE CZIP WITH Xczip, HADDRESS WITH Xaddress
REPLACE HPHONE WITH Xphone, HCITY WITH Xcity
REPLACE HSTATE WITH Xstate, HZIP WITH Xzip
REPLACE ADATE WITH CTOD(Xadate), ATIME WITH Xatime
REPLACE DDATE WITH CTOD(Xddate), DTIME WITH Xdtime
REPLACE PSIZE WITH Xpsize, CHILDREN WITH Xchildren
REPLACE PETS WITH Xpets, CRIBS WITH Xcribs
REPLACE ROLLAWAY WITH Xrollaway, HCHAIR WITH Xchair
REPLACE BNAME WITH Xbname, BADDRESS WITH Xbaddess
REPLACE BCITY WITH Xbcity, BSTATE WITH Xbstate
REPLACE BZIP WITH Xbzip, CNNAME WITH Xcnname
REPLACE CNADDRESS WITH Xcnaddress, CNCITY WITH Xcnocity
REPLACE CNSTATE WITH Xcnstate, CNZIP WITH Xcnzip
REPLACE WHOMADE WITH Xmade, WHOPHONE WITH Xmadephone
REPLACE TRAVEL_AGN WITH Xagent, WHO_TOOK WITH rsvname
REPLACE DEPST_REQ WITH Xunitdep, DEPST_RECV WITH Xdeprecvd
REPLACE DEPST_DATE WITH DATE(), DEPST_HOW WITH Xdephow

* Store Reservation Dates With Unit
USE unitres INDEX unitresu, unitresr
APPEND BLANK
REPLACE RES_NUMBER WITH rnum
REPLACE START_DATE WITH CTOD(Xadate)
REPLACE END_DATE WITH CTOD(Xddate)
REPLACE U_NUM WITH unitnum

* Update number of days unit is used
USE unit INDEX unitunit
FIND &unitnum
REPLACE U_DAYS WITH U_DAYS + (CTOD(Xddate) - CTOD(Xadate))

USE
RELEASE rnum

RETURN
PARAMETERS Xadate, Xddate
PRIVATE temp
temp = ""
DO WHILE .T.
   @ 23,1 SAY " Check-In Date:   "
   @ 23,40 SAY " Check-Out Date:   "
   @ 23,17 GET Xadate PICTURE "99/99/99"
   @ 23,48 GET Xddate PICTURE "99/99/99"
   READ
   IF CTOD(Xadate) > CTOD(Xddate)
      ? chr(7)
      @ 23,1 SAY " ?? Cannot Depart before Arrival... Press RETURN Key to Reenter! " GET temp
   READ
   ELSE
      IF Xadate = Xddate
         ? chr(7)
         @ 23,1 SAY " ?? Departure equals Arrival... Press RETURN Key to Reenter! " GET temp
         READ
      ELSE
         IF CTOD(Xadate) < DATE()
            ? chr(7)
            @ 23,1 SAY " ?? Cannot Arrive Before TODAY... Press RETURN Key to Reenter! " GET temp
         READ
         ELSE
            * Got GOOD DATES!!
            EXIT
         ENDDIF
      ENDDIF
   ENDDO
RETURN
PARAMETERS unitp, udep, urate

USE udata INDEX udatatyp

FIND &unitp

udep = U_DEPOSIT
urate = U_NIGHTRAT

USE

RETURN
* Program...: RSVSCRN1.PRG - Display Reservation Screen
* Author....: Warren M. Bartlett
* Date......: 07/15/84
* Notice....: Copyright 1984, All Rights Reserved
* Notes.....: Displays the first Reservation Screen
* Reserved.: NONE
*
CLEAR
@ 1,0 SAY "|=================================================
|=====
@ 1,55 SAY "|=================================================
@ 2,0 SAY "| The Reservation System | Screen 1
| MODE:
@ 2,55 SAY "Enter a New Reservation"
@ 2,78 SAY "|
@ 3,0 SAY "|=================================================
|=====
@ 3,55 SAY "|=================================================
@ 4,0 SAY "|=================================================

@ 4,55 SAY "| Last Name:" 
@ 5,0 SAY "| First Name:" 
@ 5,39 SAY "| MI:" 
@ 5,68 SAY "| Group Name:" 
@ 6,0 SAY "| Company:" 
@ 5,0 SAY "| Company Phone:" 
@ 5,39 SAY "| Co. Address:" 
@ 5,78 SAY "| City:" 
@ 6,0 SAY "| State:" 
@ 6,78 SAY "| Zip:" 

@ 7,0 SAY "| Home Address:" 
@ 8,0 SAY "| Home Phone:" 
@ 8,49 SAY "| City:" 
@ 8,78 SAY "| State:" 
@ 9,0 SAY "| Zip:" 
@ 9,78 SAY "|"

@ 10,0 SAY "| Company:" 
@ 10,37 SAY "| State:" 
@ 10,50 SAY "| Zip:" 
@ 10,78 SAY "|

@ 11,0 SAY "|=================================================
|=====
@ 11,55 SAY "|=================================================
@ 12,0 SAY "| Home Address:" 
@ 12,52 SAY "| Home Phone:" 
@ 12,78 SAY "|
@ 13,0 SAY "| City:" 
@ 13,37 SAY "| State:" 
@ 13,50 SAY "| Zip:" 
@ 13,78 SAY "|
@ 14,0 SAY "|================================================|
@ 14,55 SAY "---------------------------|"
@ 15,0 SAY "| Arrival:" 
@ 15,23 SAY "Time:" 
@ 15,39 SAY "| Departure:" 
@ 15,63 SAY "Time:" 
@ 15,78 SAY "|
@ 16,0 SAY "|-------------------------------|

@ 16,55 SAY "---------------------------|"
@ 17,0 SAY "| Size of Party:" 
@ 17,28 SAY "Children:" 
@ 17,53 SAY "Pets (Y or N):" 
@ 17,78 SAY "|

@ 18,0 SAY "| Number of Cribs:" 
@ 18,28 SAY "Roll-a-ways:" 
@ 18,54 SAY "High Chairs:" 
@ 18,78 SAY "|

@ 19,0 SAY "| Misc. Equipment Needed"
@ 19,78 SAY "|
@ 20,0 SAY "|=================================================================

@ 20,0 SAY "|=================================================================

@ 20,55 SAY "------------------------------------------|

RETURN
* Program..: Clear Temporary Variables (CLEARVAR)
* Author..: Warren M. Bartlett
* Date.....: 07/17/84
* Notice...: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls....: NONE
* Called by:

* PARAMETERS None
*
* Clear all variables
*
X1name=""
Xfname=""
Xmi=""
Xgname="None"
Xcompany="None"
XCphone="( )" - ""
Xcadd=""
Xccity=""
Xcstate=""
Xczip=""
Xaddress="" - ""
Xphone="( )" - ""
Xcity=""
Xstate=""
Xzip=""
* Xadate="/ / Got it Already!
Xatime="" :
* Xddate="/ / Got it Already!
Xdtime="" :
Xpsize="1"
Xchildren="0"
Xpets="N"
Xcribs="0"
Xrollaway="0"
Xchair="0"
Xequip="None"
Xbname="Same"
Xbadress=""
Xbcity=""
Xbstate=""
Xbzip=""
Xcnnname="Same"
Xcmadress=""
Xcncity=""
Xcnstate=""
Xcnzip=""
Xmade="Same"
Xmadephone="( )" - ""
Xagent="None"
Xunitdep=unitdep
Xdeprecvd=0
Xdeephow=""

RETURN
* Program...: RGETDAT2.PRG - Get values from res screen
* Author....: Warren M. Bartlett
* Date......: 07/15/84
* Notice....: Copyright 1984, All Rights Reserved
* Notes.....: Gets variables in the Reservation Screen

@ 5,21 GET Xbname PICTURE "!XXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 6,21 GET Xbaddress PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 7,21 GET Xbcity PICTURE "!XXXXXXXXXXXXXX"
@ 7,49 GET Xbstate PICTURE "!!"
@ 7,64 GET Xbzip PICTURE "99999-9999"
@ 9,34 GET Xcnnname PICTURE "!XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 10,21 GET Xcnadress PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 11,21 GET Xcnname PICTURE "!XXXXXXXXXXXXXX"
@ 11,49 GET Xcnstate PICTURE "!!"
@ 11,64 GET Xcnzip PICTURE "99999-9999"
@ 14,22 GET Xmade PICTURE "!XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 15,62 GET Xmadephone PICTURE "(999)999-9999"
@ 15,22 GET Xagent PICTURE "!XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 17,43 GET Xunitdep PICTURE "999.99"
@ 18,21 GET Xdeprecvd PICTURE "999.99"
@ 18,55 GET Xdephow PICTURE "!!"
READ
RETURN
* Program...: RSVSCRN2.PRG - Display Reservation Screen2
* Author....: Warren M. Bartlett
* Date.......: 07/15/84
* Notice....: Copyright 1984, All Rights Reserved
* Notes.....: Displays the Screen
* Reserved.: NONE
*
CLEAR
@ 1,0 SAY "|================================================================================|
   |                                        |
@ 1,55 SAY "================================================================================|
@ 2,0 SAY "I The Reservation System I Screen 2 |
   | MODE:" |
@ 2,55 SAY "Enter a New Reservation|
@ 3,0 SAY "|================================================================================|
   |                                        |
@ 3,55 SAY "================================================================================|
@ 4,0 SAY "|================================================================================|
   |                                        |
@ 4,55 SAY "================================================================================|
@ 5,0 SAY "I Bill To: Name:"
@ 5,78 SAY "|"
@ 6,0 SAY "|   Address:"
@ 6,78 SAY "|"
@ 7,0 SAY "|   City:"
@ 7,41 SAY "State:" |
@ 7,58 SAY "Zip:" |
@ 7,78 SAY "|"
@ 8,0 SAY "|================================================================================|
   |                                        |
@ 8,55 SAY "================================================================================|
@ 9,0 SAY "I Mail Confirmation To: Name:"
@ 9,78 SAY "|"
@ 10,0 SAY "|   Address:"
@ 10,78 SAY "|"
@ 11,0 SAY "|   City:"
@ 11,41 SAY "State:" |
@ 11,58 SAY "Zip:" |
@ 11,78 SAY "|"
@ 12,0 SAY "|================================================================================|
   |                                        |
@ 12,55 SAY "================================================================================|
@ 13,0 SAY "|================================================================================|
   |                                        |
@ 13,55 SAY "================================================================================|
@ 14,0 SAY "I Person Making Res:"
@ 14,78 SAY "|"
@ 15,0 SAY "I Travel Agent Name:"
@ 15,53 SAY "Phone #: "
@ 15,78 SAY "|"
I Deposit Information: Deposit Required:

I Deposit Received:

How Paid:

RETURN
* Program...: RGETDAT1.PRG - Get values from reservation
* Author....: Warren M. Bartlett
* Date......: 07/15/84
* Notice....: Copyright 1984, All Rights Reserved
* Notes.....: Gets variables in the Reservation Screen
*
set talk off
set echo OFF
@ 5,52 GET Xfname PICTURE "!AAAAAAAAAAAAAAAA"
@ 5,73 GET Xmi PICTURE "!
@ 6,15 GET Xgname PICTURE "!XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 8,12 GET Xcompany PICTURE "!XXXXXXXXXXXXXXXXXXXXXXXXX"
@ 8,64 GET XPhone PICTURE "(999)999-9999"
@ 9,17 GET Xcadd PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 10,9 GET Xcity PICTURE "!XXXXXXXXXXXXXXXXX"
@ 10,45 GET Xcstate PICTURE "!!"
@ 10,56 GET Xczip PICTURE "99999-9999"
@ 12,17 GET Xaddress PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 12,64 GET Xphone PICTURE "(999)999-9999"
@ 13,9 GET Xcality PICTURE "!XXXXXXXXXXXXXX"
@ 13,45 GET Xstate PICTURE "!!"
@ 13,56 GET Xzip PICTURE "99999-9999"
@ 15,12 SAY Xadate PICTURE "99/99/99"
@ 15,30 GET Xatime PICTURE "99:99"
@ 15,53 SAY Xddate PICTURE "99/99/99"
@ 15,70 GET Xdtime PICTURE "99:99"
@ 17,20 GET Xpsize PICTURE "99"
@ 17,42 GET Xchildren PICTURE "99"
@ 17,69 GET Xpets PICTURE "!
@ 18,20 GET Xcribs PICTURE "9"
@ 18,42 GET Xrollaway PICTURE "9"
@ 18,69 GET Xchair PICTURE "9"
@ 19,26 GET Xequip
READ
RETURN
* Program..: UNITFIND.PRG - Find the Unit for the Reservation
* Author...: Warren M. Bartlett
* Date.....: 07/17/84
* Notice...: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Called by: unitget
*
PARAMETERS unitnum, unitdesc, unittype, startdate, enddate
PRIVATE available, usedday, sdate, edate
sdate = CTOD(startdate)
edate = CTOD(enddate)
unitnum = "0"

SELECT 1
USE unit INDEX utype
SELECT 2
USE unitres index unitresu
SELECT 1
FIND &unittype

IF .NOT. EOF()
    usedday = 1000
    DO WHILE unit->U_TYPE = unittype
        IF unit->U_DAYS < usedday
* Check for availability during Time of Stay
            available = .T.
        SELECT 2
            uuuu = unit->U_NUM
            FIND &uuuu
            DO WHILE .NOT. EOF()
                IF (sdate >= START_DATE .AND. sdate < END_DATE)
                    .OR. (edate <= END_DATE .AND. edate > START_DATE)
                        available = .F.
                ENDDO
            ENDDO
        EXIT
    ENDDO
ENDIF
SKIP
IF (unit->U_NUM <> unitres->U_NUM)
    .OR. (.NOT. available)
    EXIT
ENDIF
ENDDO
SELECT 1
IF available
    unitnum = unit->U_NUM
    unitdesc = unit->U_DESCRIP
    usedday = unit->U_DAYS
ENDIF
ENDIF
SKIP
ENDDO
ENDIF
SELECT 2
USE
SELECT 1
USE
RETURN
PARAMETERS rsnum
PRIVATE cclname, ccfname, ccmi, cctp, ccnum, ccex

READ
SELECT 10
USE credited INDEX ccresnum
APPEND BLANK
REPLACE RES_NUMBER WITH rsnum
CLEAR

cclname = " "
ccfname = " "
cmi = " "
cctp = " "
ccnum = " 
ccex = " / / "
@ 10,10 SAY "Enter Credit Card Information"
@ 12,10 SAY "Last Name:"
   GET cclname PICTURE "!XXXXXXXXXXXXXXXXXXXXXX"
@ 13,10 SAY "First Name:"
   GET ccfname PICTURE "!XXXXXXXXXXXXXXXXXX"
@ 14,10 SAY "Middle Init:"
   GET cmi PICTURE "!"
@ 15,10 SAY "Credit Card Type:"
   GET cctp PICTURE "!!"
@ 16,10 SAY "Number:"
   GET ccnum PICTURE "XXXXXXXXXXXXXXXXXX" 
@ 17,10 SAY "Expiration Date:" 
   GET ccex PICTURE "99/99/99"
READ

REPLACE LNAME WITH cclname, FNAME WITH ccfname
REPLACE MI WITH cmi, CC_TYPE WITH cctp
REPLACE CC_NUMBER WITH ccnum, CC_EXPIR WITH CTOD(ccex)

USE
SELECT 1
RETURN
* Program...: Display Unit Data at Bottom (DISPUNIT)
* Author....: Warren M. Bartlett
* Date......: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Notes.....:
* Reserved.: NONE
* Calls....: NONE
* Called by: raccept
* PARAMETERS None

* Display Unit Data at Bottom of Screen
  @ 21,0 SAY "|====================================================================|
  @ 21,55 SAY "|"  
  @ 22,0 SAY "| Unit Number:"  
  @ 22,16 SAY unitnum  
  @ 22,52 SAY "Nightly Rate"  
  @ 22,68 SAY unitrate PICTURE '#999.99'  
  @ 22,78 SAY "|"  
  @ 23,0 SAY "| Features: "  
  @ 23,12 SAY unitdesc  
  @ 23,52 SAY "Deposit Required:"  
  @ 23,70 SAY unitdep PICTURE '#999.99'  
  @ 23,78 SAY "|"  
  @ 24,0 SAY "|====================================================================|
  @ 24,55 SAY "|"  
RETURN
* Program...: Modify Existing Reservation (MODIFRES)
* Author....: Warren M. Bartlett
* Date......: 07/27/84
* Notice....: Copyright 1984, All Rights Reserved
*
* PARAMETERS None
PRIVATE answer
DO WHILE .T.
  number = " "
  name = " "
  DO getresky WITH number, name
  IF name = " "
    IF number = " "
      RETURN
    ELSE
      USE reservat INDEX resnum
      FIND &number
      ENDF
    ELSE
      USE reservat INDEX resname
      FIND &name
      ENDF
  ENDIF
  IF EOF()
    answer = " "
    @ 22,2 SAY "Reservation does NOT exist... Press Enter
    Key to continue"
    @ 22,65 GET answer READ
  ELSE
    DO copyvars
    DO dispres
      answer = " "
      @ 24,3 SAY "<E>dit or <C>ancel (Enter to return)?"
      @ 24,55 GET answer PICTURE "!"
      READ
      IF answer = "E"
        DO editres
      ELSE
        IF answer = "C"
          DO rescancel
        ELSE
          EXIT
        ENDF
      ENDF
    ENDDO
  ENDDO
USE
RETURN
* Program...: Get the Key into the Reservation File (GETRESKY)
* Author....: Warren M. Bartlett
* Date......: 07/27/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls....: NONE
*
PARAMETERS number, name

CLEAR
@ 5,1 SAY "Enter Reservation Key Data"
@ 6,1 SAY "Reservation Name or Number must be entered."
@ 7,1 SAY "Press ENTER key twice to Return to The Main Menu"
name = "" ""
number = "" ""

@ 10,10 SAY "Reservation Name: "
GET name PICTURE "!AXXXXXXXXXXXXXXXXXX"
@ 11,10 SAY "Number: "
GET number PICTURE "XXXXXXXXXXXXX"

READ

RETURN
* Program...: Edit Credit Card Info (EDITCC)
* Author....: Warren M. Bartlett
* Date......: 07/17/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls.....: NONE
*
* PARAMETERS none

PRIVATE cclname, ccfname, ccmi, cctp, ccnum, ccex

USE credited INDEX ccresnum
FIND &Xrnum

CLEAR
cclname = LNAME
ccfname = FNAME
ccmi = MI
ccctp = CC_TYPE
ccnum = CC_NUMBER
ccex = DTOC(CC_EXPIR)
@ 10,10 SAY "Edit Credit Card Information "
@ 12,10 SAY "Last Name:
   GET cclname PICTURE "!XXXXXXXXXXXXXXXXXX"
@ 13,10 SAY "First Name:
   GET ccfname PICTURE "!XXXXXXXXXXXXXXXX"
@ 14,10 SAY "Middle Init:
   GET ccmi PICTURE "!
@ 15,10 SAY "Credit Card Type:
   GET cctp PICTURE "!!
@ 16,10 SAY "Number:
   GET ccnum PICTURE "XXXXXXXXXXXXXXXXXXXX"
@ 17,10 SAY "Expiration Date:
   GET ccex PICTURE "99/99/99"

READ

REPLACE LNAME WITH cclname, FNAME WITH ccfname
REPLACE MI WITH ccmi, CC_TYPE WITH cctp
REPLACE CC_NUMBER WITH ccnum, CC_EXPIR WITH CTOD(ccex)

USE
SELECT 1
RETURN
* Program...: Edit Reservation (EDITRES)
* Author....: Warren M. Bartlett
* Date......: 07/27/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls.....: NONE
*

@ 2,55 SAY "Edit Reservation
@ 5,14 GET Xlname PICTURE "XXXXXXXXXXXXXXXXXXXX"  
@ 5,52 GET Xfname PICTURE "!AAAAAAAAAAAAAAA"  
@ 5,73 GET Xmip PICTURE "!"  
@ 6,15 GET Xgname PICTURE "!XXXXXXXXXXXXXXXXXXXXXXXXX"  
@ 8,12 GET Xcompany PICTURE "!XXXXXXXXXXXXXXXXXXXXXXXXX"  
@ 8,64 GET Xphone PICTURE "(999)999-9999"  
@ 9,17 GET Xcadd PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"  
@ 10,9 GET Xccity PICTURE "!XXXXXXXXXXXXXXXXX"  
@ 10,45 GET Xstate PICTURE "!!"  
@ 10,56 GET Xzipcode PICTURE "99999-9999"  
@ 12,17 GET Xaddress PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXX"  
@ 12,64 GET Xphone PICTURE "(999)999-9999"  
@ 13,9 GET Xcity PICTURE "!XXXXXXXXXXXXXXXXX"  
@ 13,45 GET Xstate PICTURE "!!"  
@ 13,56 GET Xzip PICTURE "99999-9999"  
@ 15,12 GET Xadate PICTURE "99/99/99"  
@ 15,30 GET Xdate PICTURE "99/99/99"  
@ 15,53 GET Xddate PICTURE "99/99/99"  
@ 15,70 GET Xddate PICTURE "99:99"  
@ 17,20 GET Xsize PICTURE "99"  
@ 17,42 GET Xchildren PICTURE "99"  
@ 17,69 GET Xpets PICTURE "!"  
@ 18,20 GET Xcribs PICTURE "9"  
@ 18,42 GET Xrollaway PICTURE "9"  
@ 18,69 GET Xchair PICTURE "9"  
@ 19,26 GET Xequip  
@ 21,66 GET Xunitnum PICTURE "XXXXXXXXXXXX"  
READ  
DO dispres2  
DO editres2  
IF Xdeprecvd > 0  
   DO editcc  
ENDIF  
answer = "Y"  
@ 24,2 SAY "Do you wish to save these changes (Y or N)?"  
@ 24,55 GET answer PICTURE "!"  
READ  
IF answer = "Y"  
   DO savechng  
ENDIF  
RETURN
* Program...: Copy to Temporary Variables (COPYVARS)
* Author...: Warren M. Bartlett
* Date.....: 07/27/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls....: NONE
*
* PARAMETERS None
Xrnum = RES_NUMBER
Xrdate = DTOC(RES_DATE)
Xunitnum = RES_UNIT
Xlname = LNAME
Xfname = FNAME
Xmi = MI
Xgname = GNAME
Xcompany = COMPANY
Xccphone = CPHONE
Xcadd = CADDRESS
Xccity = CCITY
Xcstate = CSTATE
Xczip = CZIP
Xaddress = HADDRESS
Xphone = HPHONE
Xccity = HCITY
Xstate = HSTATE
Xzip = HZIP
Xadate = DTOC(ADATE)
Xatime = ATIME
Xddate = DTOC(DDATE)
Xdtime = DTIME
Xpsize = PSIZE
Xchildren = CHILDREN
Xpets = PETS
Xcribs = CRIBS
Xrollaway = ROLLAWAY
Xchchair = HCHAIR
Xbname = BNAME
Xbaddress = BADDRESS
Xbcity = BCITY
Xbstate = BSTATE
Xbzip = BZIP
Xcnname = CNNAME
Xcnadress = CNADDRESS
Xcncity = CNCITY
Xcnstate = CNSTATE
Xcnczip = CNZIP
Xmade = WHOMADE
Xmadephone = WHOPHONE
Xagent = TRAVEL_AGN
Xravname = WHO_TOOK
Xunitdep = DEPST_REQ
Xdeprecvd = DEPST_RECV
Xdepd = DTOC(DEPST_DATE)
Xdephow = DEPST_HOW
RETURN
Program..: Display Reservation (DISPRES)
Author..: Warren M. Bartlett
Date....: 07/27/84
Notice...: Copyright 1984, All Rights Reserved
Reserved.: NONE
Calls....: NONE

DO rsvscrnl
@ 2,55 SAY "Display Reservation  
@ 5,14 SAY Xlname
@ 5,52 SAY Xfname
@ 5,73 SAY Xmi
@ 6,15 SAY Xgname
@ 8,12 SAY Xcompany
@ 8,64 SAY TRIM(XCphone)
@ 9,17 SAY Xcadd
@ 10,9 SAY Xccity
@ 10,45 SAY Xcstate
@ 10,56 SAY Xczip
@ 12,17 SAY Xaddress
@ 12,64 SAY TRIM(Xphone)
@ 13,9 SAY Xcity
@ 13,45 SAY Xstate
@ 13,56 SAY Xzip
@ 15,12 SAY Xadate
@ 15,30 SAY Xtime
@ 15,53 SAY Xddate
@ 15,70 SAY Xdtime
@ 17,20 SAY Xpsize
@ 17,42 SAY Xchildren
@ 17,69 SAY Xpets
@ 18,20 SAY Xcribs
@ 18,42 SAY Xrollaway
@ 18,69 SAY Xchair
@ 19,26 SAY Xequip
@ 20,0 SAY "=====================================================================
@ 20,47 SAY "=====================================================================
@ 21,0 SAY "| RES. NUMBER:"
@ 21,20 SAY Xrnum
@ 21,52 SAY "UNIT NUMBER:"
@ 21,66 SAY Xunitnum
@ 21,78 SAY "|
@ 22,0 SAY "| RESERVATIONIST:"
@ 22,20 SAY Xrsvname
@ 22,60 SAY "DATE: 
@ 22,66 SAY Xrdate
@ 22,78 SAY "|
@ 23,0 SAY "|=====================================================================
@ 23,47 SAY "=====================================================================
RETURN
* Program: Display Reservation - Screen 2 (DISPRES2)
* Author: Warren M. Bartlett
* Date: 07/27/84
* Notice: Copyright 1984, All Rights Reserved
* Calls: NONE

DO rsvscrn2

@ 2,55 SAY "Display Reservation"

@ 20,0 SAY "|======================================================================|

@ 20,55 SAY "|"  

* Display Reservation Number and Date at Bottom of Screen

@ 21,0 SAY "| RES. NUMBER:"  

@ 21,20 SAY Xrnum

@ 21,52 SAY "UNIT NUMBER:"  

@ 21,66 SAY Xunitnum

@ 21,78 SAY "|

@ 22,0 SAY "| RESERVATIONIST:"  

@ 22,20 SAY Xrsvname

@ 22,60 SAY "DATE: "  

@ 22,66 SAY Xrdate

@ 22,78 SAY "|

@ 23,0 SAY "|======================================================================|

@ 23,55 SAY "|"

RETURN
* Program.: Edit Reservation - Screen 2 (EDITRES2)
* Author....: Warren M. Bartlett
* Date......: 07/27/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls....: NONE
*
@ 2,55 SAY "Edit Reservation  
  
@ 5,21 GET Xbname PICTURE "!XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 6,21 GET Xbadress PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXX"
@ 7,21 GET Xbcity PICTURE "!XXXXXXXXXXXXXX"
@ 7,49 GET Xbstate PICTURE "!!"
@ 7,64 GET Xbzip PICTURE "99999-9999"
@ 9,34 GET Xcname PICTURE "!XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 10,21 GET Xcnadress PICTURE "XXXXXXXXXXXXXXXXXXXXXXX"
@ 11,21 GET Xcnicity PICTURE "!XXXXXXXXXXXXXX"
@ 11,49 GET Xcnstate PICTURE "!!"
@ 11,64 GET Xcnzip PICTURE "99999-9999"
@ 14,22 GET Xmade PICTURE "!XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 15,62 GET Xmadephone PICTURE "(999)999-9999"
@ 15,22 GET Xagent PICTURE "!XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 17,43 GET Xunitdep PICTURE "999.99"
@ 18,21 GET Xdeprecvd PICTURE "999.99"
@ 18,55 GET Xdephow PICTURE "!!"
READ
RETURN
* Program...: Save Changes (SAVECHNG)
* Author....: Warren M. Bartlett
* Date......: 07/27/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls.....: NONE
*
* PARAMETERS None
*
REPLACE RES_NUMBER WITH Xrnum
REPLACE RES_DATE WITH CTOD(Xrdate)
REPLACE RES_UNIT WITH Xunitnum
REPLACE LNAME WITH Xlname
REPLACE FNAME WITH Xfname
REPLACE MI WITH Xmi
REPLACE GNAME WITH Xgname
REPLACE COMPANY WITH Xcompany
REPLACE CPHONE WITH XCphone
REPLACE CADDRESS WITH Xcadd
REPLACE CCITY WITH Xccity
REPLACE CSTATE WITH Xcstate
REPLACE CZIP WITH Xczip
REPLACE HADDRESS WITH Xaddress
REPLACE HPHONE WITH Xphone
REPLACE HCITY WITH Xcity
REPLACE HSTATE WITH Xstate
REPLACE HZIP WITH Xzip
REPLACE ADATE WITH CTOD(Xadate)
REPLACE ATIME WITH Xatime
REPLACE DDATE WITH CTOD(Xddate)
REPLACE DTIME WITH Xdtime
REPLACE PSIZE WITH Xpsize
REPLACE CHILDREN WITH Xchildren
REPLACE PETS WITH Xpets
REPLACE CRIBS WITH Xcribs
REPLACE ROLLAWAY WITH Xrollaway
REPLACE RCHAIR WITH Xhchair
REPLACE BNAME WITH Xbname
REPLACE BADDRESS WITH Xbadress
REPLACE BCITY WITH Xbcity
REPLACE BSTATE WITH Xbstate
REPLACE BZIP WITH Xbzip
REPLACE CNNAME WITH Xcnnname
REPLACE CNADDRESS WITH Xcnaddress
REPLACE CNCYCITY WITH Xcnr
city
REPLACE CNSTATE WITH Xcnstate
REPLACE CNZIP WITH Xcnzip
REPLACE WHOMADE WITH Xmade
REPLACE WHOPHONE WITH Xmadephone
REPLACE TRAVEL_AGN WITH Xagent
REPLACE WHO_TOOK WITH Xrsvname
REPLACE DEPST_REQ WITH Xunitdep
REPLACE DEPST_RECV WITH Xdeprecvd
REPLACE DEPST_DATE WITH CTOD(Xdepdate)
REPLACE DEPST_HOW WITH Xdephow
RETURN
Program..: Display Deposit Status Screen (DEPScreen)
Author...: Warren M. Bartlett
Date......: 07/27/84
Notice....: Copyright 1984, All Rights Reserved
Reserved.: NONE
Calls.....: NONE
Called by: dispdep

@ 1,0 SAY "|==================================|
|-------|
@ 1,55 SAY "------------------------------|
@ 2,0 SAY "| The Reservation System |
| MODE:
@ 2,55 SAY "Maintain Deposit |
@ 3,0 SAY "|-------------------------------|
|-------|
@ 3,55 SAY "------------------------------|
@ 4,0 SAY "==================================

@ 4,55 SAY "------------------------------|
@ 5,0 SAY "| Reservation Number:|
@ 5,78 SAY "|
@ 6,0 SAY "| Reservation Name:|
@ 6,78 SAY "|
@ 7,0 SAY "| Home Address:|
@ 7,78 SAY "|
@ 8,0 SAY "| City:|
@ 8,39 SAY "State:"
@ 8,55 SAY "Zip Code:"
@ 8,78 SAY "|
@ 9,0 SAY "|-------------------------------|
|-------|
@ 9,55 SAY "------------------------------|
@ 10,0 SAY "| Company Name:|
@ 10,78 SAY "|
@ 11,0 SAY "| Address:"
@ 11,78 SAY "|
@ 12,0 SAY "| City:"
@ 12,39 SAY "State:"
@ 12,55 SAY "Zip Code:"
@ 12,78 SAY "|
@ 13,0 SAY "|-------------------------------|
|-------|
@ 13,55 SAY "------------------------------|
@ 14,0 SAY "| Deposit Required:"
@ 14,78 SAY "|
@ 15,0 SAY "| Deposit Received:"
@ 15,42 SAY "Date Received:"
@ 15,78 SAY "|
@ 16,0 SAY "| How Paid:"
I Credit Card Information

| CC Type: |
| CC Number: |
| Expires: |

RETURN
PRIVATE rnum

rnum = RES_NUMBER

@ 5,23 SAY RES_NUMBER
@ 6,23 SAY FNAME
@ 6,31 SAY LNAME
@ 7,23 SAY HADDRESS
@ 8,10 SAY HCITY
@ 8,48 SAY HSTATE
@ 8,66 SAY HZIP
@ 10,17 SAY COMPANY
@ 11,17 SAY CADDRESS
@ 12,10 SAY CCITY
@ 12,48 SAY CSTATE
@ 12,66 SAY CZIP
@ 14,21 SAY DEPST_REQ
@ 15,21 SAY DEPST_RECV
@ 15,58 SAY DEPST_DATE
@ 16,13 SAY DEPST_HOW

SELECT 2
USE credited INDEX ccresnum
FIND &rnum
IF .NOT. EOF()
   @ 19,12 SAY CC_TYPE
   @ 20,14 SAY CC_NUMBER
   @ 20,59 SAY CC_EXPIR
ELSE
   @ 19,12 SAY "NONE"
ENDIF
SELECT 1
RETURN
* Program...: Maintain Deposit Status (DEPMAINT)
* Author...: Warren M. Bartlett
* Date.....: 07/27/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls....: NONE
* Called by: Menu

DO WHILE .T.
    number = " "
    name = " 
    DO getresky WITH number, name
    IF name = " "
       IF number = " "
          RETURN
       ELSE
          USE reservat INDEX resnum
          FIND &number
          ENDF
    ELSE
       USE reservat INDEX resname
       FIND &name
       ENDF
    IF EOF()
       answer = " "
       @ 22,2 SAY "Reservation does NOT exist... Press Enter Key to continue"
       @ 22,65 GET answer
       READ
       ELSE
          CLEAR
          DO depscrn
          DO dispdep
          answer = " 
          @ 24,3 SAY "<E>d it or <C>ancel (Enter to return)?"
          @ 24,55 GET answer PICTURE "!"
          READ
          IF answer = "E"
             DO editdep
          ELSE
             IF answer = "C"
                DO rescancel
             ELSE
                EXIT
             ENDF
          ENDF
    ENDF
ENDO
USE
ENDDO
USE
RETURN
PRIVATE Xunitdep, Xdeprecvd, Xdepdate, Xdephow
PRIVATE Xccctype, Xccnumber, Xccexp, answer

Xunitdep = DEPST_REQ
Xdeprecvd = DEPST_RECV
Xdepdate = DTOC(DEPST_DATE)
Xdephow = DEPST_HOW
@ 14,21 GET Xunitdep PICTURE "999.99"
@ 15,21 GET Xdeprecvd PICTURE "999.99"
@ 15,58 GET Xdepdate PICTURE "99/99/99"
@ 16,13 GET Xdephow PICTURE "!!"
READ
SELECT 2
IF .NOT. EOF()
   Xccctype = CC_TYPE
   Xccnumber = CC_NUMBER
   Xccexp = DTOC(CC_EXPIR)
ELSE
   Xccctype = " "
   Xccnumber = " "/"
   Xccexp = " "/"
ENDIF
@ 19,12 GET Xccctype PICTURE "!!"
@ 20,14 GET Xccnumber
@ 20,59 GET Xccexp PICTURE "99/99/99"
READ

answer = " Y"
@ 24,2 SAY "Do you wish to save these changes (Y or N) ?"
@ 24,55 GET answer PICTURE "!!"
READ
IF answer = "Y"
   IF Xccctype <> " "
      IF EOF()
         APPEND BLANK
         REPLACE CREDITCD->RES_NUMBER WITH RESERVAT->RES_NUMBER
         REPLACE CREDITCD->LNAME WITH RESERVAT->LNAME
         REPLACE CREDITCD->FNAME WITH RESERVAT->FNAME
         REPLACE CREDITCD->MI WITH RESERVAT->MI
      ENDIF
      REPLACE CC_TYPE WITH Xccctype, CC_NUMBER WITH Xccnumber
   ELSE
      IF Xccctype <> " "
         REPLACE CREDITCD->RES_NUMBER WITH RESERVAT->RES_NUMBER
         REPLACE CREDITCD->LNAME WITH RESERVAT->LNAME
         REPLACE CREDITCD->FNAME WITH RESERVAT->FNAME
         REPLACE CREDITCD->MI WITH RESERVAT->MI
      ENDIF
      REPLACE CC_TYPE WITH Xccctype, CC_NUMBER WITH Xccnumber
      IF .NOT. EOF()
         APPEND BLANK
         REPLACE CREDITCD->RES_NUMBER WITH RESERVAT->RES_NUMBER
         REPLACE CREDITCD->LNAME WITH RESERVAT->LNAME
         REPLACE CREDITCD->FNAME WITH RESERVAT->FNAME
         REPLACE CREDITCD->MI WITH RESERVAT->MI
      ENDIF
      REPLACE CC_TYPE WITH Xccctype, CC_NUMBER WITH Xccnumber
   ENDIF
ENDIF
REPLACE CC_EXPIR WITH CTOD(Xccexp)
ENDIF
USE
SELECT 1
REPLACE DEPST_REQ WITH Xunitdep
REPLACE DEPST_RECV WITH Xdeprecvd
REPLACE DEPST_DATE WITH CTOD(Xdepdate)
REPLACE DEPST_HOW WITH Xdephow
ELSE
USE
SELECT 1
ENDIF
RETURN
* Program...: Generate Reports Menu (REPRTGEN)
* Author....: Warren M. Bartlett
* Date......: 07/27/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls.....: NONE
* Called by: menu
*
* PARAMETERS None
*
* Get Menu Selection
DO WHILE .T.
   CLEAR
   @ 1,25 SAY "|=================================|
   @ 2,25 SAY "| The Reservation System |
   @ 3,25 SAY "| Copyright (c) 1984 |
   @ 4,25 SAY "| Warren M. Bartlett |
   @ 5,25 SAY "|=================================|
   @ 6,16 SAY "|=================================|
   @ 7,16 SAY "| Report Menu |
   @ 8,16 SAY "|=================================|
   @ 9,16 SAY "|
   @ 10,16 SAY "| 0. EXIT |
   @ 11,16 SAY "| 1. Cancellation Report |
   @ 12,16 SAY "| 2. Arrival/Departure Report |
   @ 13,16 SAY "| 3. Reservation Status Report |
   @ 14,16 SAY "| 4. Usage Report |
   @ 15,16 SAY "|
   @ 16,16 SAY "|
   @ 17,16 SAY "| Enter Your Choice: |
   @ 18,16 SAY "|
   @ 19,16 SAY "|=================================|

STORE 7 TO selectnum
DO WHILE selectnum < 0 .OR. selectnum > 6
   STORE "0" TO select
   @ 17,40 GET select PICTURE "9"
   READ
   STORE VAL(select) TO selectnum
ENDDO
DO CASE
CASE selectnum= 0
  RETURN
CASE selectnum= 1
  * DO 1. Cancellation Report
  DO cancrept
CASE selectnum= 2
  * DO 2. Arrival/Departure Report
  DO aanddrpt
CASE selectnum= 3
  * DO 3. Reservation Status Report
  DO rstatrpt
CASE selectnum= 4
  * DO 4. Usage Report
  DO userept
ENDCASE
ENDDO T

* EOF: Report Menu
* Program...: Arrival And Departure Reports (AANDDRPT)
* Author....: Warren M. Bartlett
* Date......: 07/27/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls.....: NONE
* Called By: REPRTGEN
* PARAMETERS None
PRIVATE answer, rdate, panswer, dummy
ranswer = "B"
@ 10,10 SAY "Arrival and Departure Reports"
@ 11,10 SAY "=================================="
@ 12,10 SAY "Which Reports do you wish? (<A>rrival,
<D>epart, <B>oth )"
@ 12,70 GET ranswer PICTURE ":!
READ
rdate = DTOC(DATE()+1)
@ 14,10 SAY "For What Date Would You like the Report(s)?"
@ 14,55 GET rdate PICTURE "99/99/99"
READ
panswer = "S"
@ 16,10 SAY "Report to <P>rinter or <S>creen? "
GET panswer PICTURE "":!
READ
IF ranswer = "B" .OR. ranswer = "A"
  USE reservat INDEX resname
  IF panswer = "P"
    WAIT "Check Printer... Press any Key when Ready"
    REPORT FORM arrival FOR ADATE=CTOD(rdate) NOEJECT
  ELSE
    REPORT FORM arrival FOR ADATE=CTOD(rdate)
    WAIT "Press any Key to Continue..." TO dummy
  ENDIF
ELSE
ENDIF
IF ranswer = "B" .OR. ranswer = "D"
USE guest INDEX gdstname
IF panswer = "P"
  IF ranswer = "D"
    WAIT "Check Printer... Press any Key when Ready"
  ENDIF
  REPORT FORM depart FOR DDATE=CTOD(rdate) NOEJECT
  TO PRINT
ELSE
  REPORT FORM depart FOR DDATE=CTOD(rdate)
  WAIT "Press any Key to Continue..." TO dummy
ENDIF
USE
ENDIF
RETURN
* * Program...: Do Cancellation Report (CANCREPT)
* Author....: Warren M. Bartlett
* Date......: 07/27/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls.....: NONE
* Called by: REPRTGEN
*
* PARAMETERS None

PRIVATE sdate, fdate, panswer, dummy
CLEAR
sdate = DTOC(DATE())
@ 10,10 SAY "Cancellation Report"
@ 11,10 SAY "=================================="
@ 12,10 SAY "You must specify a START and END date for this report."
@ 13,10 SAY "The report will list all cancellations between START and FINISH"
@ 14,10 SAY "Enter START Date for Report "
@ 14,38 GET sdate PICTURE "99/99/99"
READ
fdate = DTOC(DATE())
@ 16,10 SAY "Enter FINISH Date for Report "
@ 16,38 GET fdate PICTURE "99/99/99"
READ
panswer = "S"
@ 18,10 SAY "Report to <P>rinter or <S>creen? "
GET panswer PICTURE "!
READ

USE cancelld INDEX candate
IF panswer = "P"
  WAIT "Check Printer... Press any Key when Ready"
  REPORT FORM cancel FOR CAN_DATE>=CTOD(sdate) .AND.
  CAN_DATE<=CTOD(fdate) NOEJECT TO PRINT
ELSE
  REPORT FORM cancel FOR CAN_DATE>=CTOD(sdate) .AND.
  CAN_DATE<=CTOD(fdate)
  WAIT "Press any Key to Continue..." TO dummy
ENDIF
USE
RETURN
* Program...: Do Reservation Status Report (RSTATRPT)
* Author....: Warren M. Bartlett
* Date......: 07/27/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls.....: NONE
* Called by: REPRTGEN
*
* PARAMETERS None
PRIVATE sdate, fdate, panswer, dummy
CLEAR
sdate = DTOC(DATE())
@ 10,10 SAY "Reservation Status Report"
@ 11,10 SAY "=================================="
@ 12,10 SAY "You must specify a START and END date for this report."
@ 13,10 SAY "The report will list all reservations due to arrive between"
@ 13,61 SAY "START and FINISH"
@ 14,10 SAY "Enter START Date for Report ", sdate PICTURE "99/99/99"
READ
fdate = DTOC(DATE())
@ 16,10 SAY "Enter FINISH Date for Report ", fdate PICTURE "99/99/99"
READ
panswer = "S"
@ 18,10 SAY "Report to <P>rinter or <S>creen? ", panswer PICTURE "!"
READ
USE reservat INDEX resname
IF panswer = "P"
    WAIT "Check Printer... Press any Key when Ready"
    REPORT FORM resrpt FOR ADATE>=CTOD(sdate) .AND. ADATE<=CTOD(fdate) NOEJECT TO PRINT
ELSE
    REPORT FORM resrpt FOR ADATE>=CTOD(sdate) .AND. ADATE<=CTOD(fdate)
    WAIT "Press any Key to Continue..." TO dummy
ENDIF
USE
RETURN
* Program...: File Maintenance (FILMAINT)
* Author....: Warren M. Bartlett
* Date......: 07/29/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls.....: NONE
* Called by: menu
* PARAMETERS None
PRIVATE select, selectnum

* Get Menu Selection
DO WHILE .T.
   CLEAR
   @ 1,25 SAY "|====================================|
   @ 2,25 SAY "| The Reservation System     |
   @ 3,25 SAY "| Copyright (c) 1984       |
   @ 4,25 SAY "| Warren M. Bartlett      |
   @ 5,25 SAY "|====================================|
   @ 6,16 SAY "|====================================|
   @ 7,16 SAY "| File Maintenance Menu     |
   @ 8,16 SAY "|====================================|
   @ 9,16 SAY "|
   @ 10,16 SAY "| 0. EXIT                    |
   @ 11,16 SAY "| 1. Backup Data Files      |
   @ 12,16 SAY "| 2. Add Arrivals to Guest File|
   @ 13,16 SAY "| 3. Maintain Unit Data File |
   @ 14,16 SAY "| 4. Purge Files            |
   @ 15,16 SAY "|
   @ 16,16 SAY "|
   @ 17,16 SAY "| Enter Your Choice:         |
   @ 18,16 SAY "|
   @ 19,16 SAY "|====================================|

STORE 5 TO selectnum
DO WHILE selectnum < 0 .OR. selectnum > 4
   STORE "0" TO select
   @ 17,40 GET select PICTURE "9"
   READ
STORE VAL(select) TO selectnum
ENDDO

DO CASE
  CASE selectnum= 0
    RETURN
  CASE selectnum= 1
    * DO 1. Backup Data Files
    DO backup
  CASE selectnum= 2
    * DO 2. Add Arrivals to Guest File
    DO addarrvl
  CASE selectnum= 3
    * DO 3. Maintain Unit Data File
    DO unitmain
  CASE selectnum= 4
    * DO 4. Purge Files
    DO purge
ENDCASE
ENDDO T

* EOF: File Maintenance Menu
* Program...: Purge Data from Files
* Author....: Warren M. Bartlett
* Date.......: 07/27/84
* Notice....: Copyright 1984, All Rights Reserved
* Notes.....: Purges from the following Files:
  * Inactive.dbf
  * Cancelled.dbf
  * Waitlist.dbf
* Reserved.: NONE
* Calls.....: NONE
* Called by: filmaint
*
* PARAMETERS
PRIVATE answer
CLEAR
answer = ""
@ 10,10 SAY "This function will REMOVE all OLD Data
(Over 90 Days old)"
@ 11,10 SAY "It Operates on the following Files:"  
@ 12,20 SAY "CANCELED - If Canceled over 90 Days ago."
@ 13,20 SAY "WAITING LIST - If Departure Date over
  90 Days ago."
@ 14,20 SAY "INACTIVE - If Checked Out 90 Days ago."
@ 23,10 SAY "DO You REALLY want to Purge this DATA? "
    GET answer PICTURE "!
READ
IF answer <> "Y"
    CLEAR
    RETURN
ENDIF

SET TALK ON
USE inactive
DELETE ALL FOR CKOUT_DATE > DATE()+90
PACK

USE waitlist
DELETE ALL FOR DDATE > DATE()+90
PACK

USE cancelled INDEX candate
DELETE ALL FOR CAN_DATE > DATE()+90
PACK

SET TALK OFF
WAIT
RETURN
* Program...: Add a unit to Unit File (UNITADD)
* Author....: Warren M. Bartlett
* Date......: 07/27/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls.....: NONE
* Called by: filmaint

* PARAMETERS None
PRIVATE Xnum,Xtype,Xdesc,Xcadd,Xphone,Xcity,Xstate,Xzip
PRIVATE Xbroom,Xbeds,Xbaths,Xlock

Xnum = " "
Xtype = " 
Xdesc = " 
Xcadd = " 
Xphone = "( ) - "
Xcity = " 
Xstate = " 
Xzip = " - "
Xbroom = "0"
Xbeds = "1"
Xbaths = "1"
Xlock = "N"

@ 2,55 SAY "Add a New Unit |"

* Get NEW Unit Number
@ 5,16 GET Xnum
READ
USE unit INDEX unitunit, utype
FIND &Xnum
 IF .NOT. EOF()
   @ 6,0 CLEAR
   ? CHR(7)
   @ 7,10 SAY "Unit: "+Xnum+" Already Exists!!"
   @ 8,10 SAY "Use EDIT function to Alter Existing Unit!"
   WAIT
   USE
   RETURN
 ENDIF
 DO disptype WITH Xtype
 @ 4,29 CLEAR
 DO addscrn
 @ 15,1 CLEAR
 @ 8,15 GET Xdesc
 @ 9,12 GET Xcadd PICTURE "XXXXXXXXXXXXXXXXXXXXXX"
 @ 9,56 GET Xphone PICTURE "(999)999-9999"
 @ 10,9 GET Xcity PICTURE "XXXXXXXXXXXXXXXX"
 @ 10,47 GET Xstate PICTURE "!!"
 @ 10,65 GET Xzip PICTURE "99999-9999"
 @ 12,13 GET Xbroom PICTURE "9"
 @ 12,38 GET Xbeds PICTURE "99"
@ 12,62 GET Xbaths PICTURE "9"
@ 13,21 GET Xlock PICTURE "X"
READ

APPEND BLANK
REPLACE U_NUM WITH Xnum
REPLACE U_TYPE WITH Xtype
REPLACE U_DESCRIP WITH Xdesc
REPLACE U_ADDR WITH Xcadd
REPLACE U_PHONE WITH Xphone
REPLACE U_CITY WITH Xcity
REPLACE U_STATE WITH Xstate
REPLACE U_ZIP WITH Xzip
REPLACE U_BEDROOMS WITH Xbroom
REPLACE U_BEDS WITH Xbeds
REPLACE U_BATHS WITH Xbaths
REPLACE U_LOCKOFF WITH Xlock
REPLACE U_DAYS WITH 0

USE
RETURN
* Program...: Add Today's Arrivals to Guest File (ADDARRVL)
* Author....: Warren M. Bartlett
* Date......: 07/27/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls.....: NONE
* Called by: filmaint
*
* PARAMETERS None

CLEAR
RESTORE FROM adddate

IF lastdate >= DATE()
  @ 10,10 SAY "Today's Arrivals HAVE ALREADY BEEN ADDED to the Guest File!"
  WAIT
  RELEASE lastdate
  RETURN
ENDIF

@ 10,10 SAY "Please Wait.... This will take a moment."

USE reservat
COPY TO arr.tmp FOR ADATE = DATE()

USE guest INDEX gname, gunit, gnum
APPEND FROM arr.tmp

ERASE arr.tmp
ERASE arr.dbt

lastdate = DATE()
SET SAFETY OFF
SAVE TO adddate ALL LIKE lastdate
SET SAFETY ON
USE

RETURN
* Program...: Backup or Restore All Data Files (BACKUP)
* Author....: Warren M. Bartlett
* Date......: 07/27/84
* Notice....: Copyright 1984, All Rights Reserved
* Reserved.: NONE
* Calls.....: NONE
* Called by: filmaint
*
* PARAMETERS None
PRIVATE answer

* Data Base Files
* WAITLIST.DBF UNIT.DBF UNITRES.DBF CREDITCD.DBF
* LOCKOFF.DBF PDATA.DBF MESSAGE.DBF UDATA.DBF
* CANCELLD.DBF GUEST.DBF INACTIVE.DBF RESERVAT.DBF

* Index Files
* CCRESCNUM.NDX UTYPE.NDX UDATATYP.NDX RESNUM.NDX
* UNITRESR.NDX UNITRESU.NDX UNITUNIT.NDX GSTNAME.NDX
* RESNAME.NDX CANDATE.NDX

* Memo Files
* WAITLIST.DBT UNIT.DBT RESERVAT.DBT MESSAGE.DBT
* CANCELLD.DBT GUEST.DBT INACTIVE.DBT

CLEAR
answer = "B"
@ 10,10 SAY "File BACKUP and RESTORE Utility"
@ 11,10 SAY "Do You want to <B>ackup or <R>estore? ",
@ 11,50 GET answer PICTURE "!"
READ

IF answer = "R"
  ? chr(7)
  answer = "N"
  @ 13,10 SAY "ARE YOU SURE? (Y or N) ",
  @ 13,34 GET answer PICTURE "!"
  READ
  IF answer = "Y"
  RUN RESTORE A: \reserv\data
  ENDIF
ELSE
  IF answer = "B"
  RUN BACKUP \reserv\data A:
  ENDIF
ENDIF
RETURN
PARAMETERS Xtype

* unit type
@ 4,29 SAY "|=============================================================================

@ 5,29 SAY "|

@ 6,29 SAY "|

@ 7,29 SAY "|

@ 8,29 SAY "|

@ 9,29 SAY "|

@ 10,29 SAY "|

@ 11,29 SAY "|

@ 12,29 SAY "|

@ 13,29 SAY "|

@ 14,29 SAY "|

@ 15,29 SAY "|

@ 16,29 SAY "|

@ 17,29 SAY "|

@ 18,29 SAY "|

@ 19,29 SAY "|

@ 20,29 SAY "|=============================================================================

SELECT 2
USE udata INDEX udatatyp
In = 5
DO WHILE .NOT. EOF()
   @ In,32 SAY udata->U_TYPE"+udata->U_DESCRIP
   ln = ln+1
   max = udata->U_TYPE
SKIP
ENDDO
USE
SELECT 1
Xtype = " "
DO WHILE (Xtype < "A" .OR. Xtype > max)
   @ 6,14 GET Xtype PICTURE "!"
   READ
ENDDO
RETURN
* Program...: Edit a unit in Unit File (EDUNIT)  
* Author....: Warren M. Bartlett  
* Date......: 08/01/84  
* Notice....: Copyright 1984, All Rights Reserved  
* Reserved..: None  
* Calls......:  
* Called by: unitmain  
*  
* PARAMETERS None  
PRIVATE Xnum,Xtype,Xdesc,Xcadd,Xphone,Xcity,Xstate,Xzip  
PRIVATE Xbroom,Xbeds,Xbaths,Xlock, answer  
*  
@ 2,55 SAY "Edit Unit Data !"  
* Get Unit Number  
Xnum = " "  
@ 5,16 GET Xnum  
READ  
USE unit INDEX unitunit, utype  
FIND &Xnum  
IF EOF()  
@ 6,0 CLEAR  
? CHR(7)  
@ 7,10 SAY "Unit: " +Xnum+ " Does NOT Exist!!"  
@ 8,10 SAY "Use ADD function to Add New Units!"  
WAIT  
USE  
RETURN  
ENDIF  
Xtype = U_TYPE  
Xdesc = U_DESCRIP  
Xcadd = U_ADDR  
Xphone = U_PHONE  
Xcity = U_CITY  
Xstate = U_STATE  
Xzip = U_ZIP  
Xbroom = U_BEDROOMS  
Xbeds = U_BEDS  
Xbaths = U_BATHS  
Xlock = U_LOCKOFF  
@ 15,1 CLEAR  
@ 5,16 SAY Xnum  
@ 6,14 GET Xtype PICTURE ".!"  
@ 8,15 GET Xdesc  
@ 9,12 GET Xcadd PICTURE "XXXXXXXXXXXXXXXXXXXXXXX"  
@ 9,56 GET Xphone PICTURE "(999)999-9999"  
@ 10,9 GET Xcity PICTURE "XXXXXXXXXXXXXXXX"  
@ 10,47 GET Xstate PICTURE ".!!"  
@ 10,65 GET Xzip PICTURE "99999-9999"  
@ 12,13 GET Xbroom PICTURE "9"  
@ 12,38 GET Xbeds PICTURE "99"  
@ 12,62 GET Xbaths PICTURE "9"
@ 13,21 GET Xlock PICTURE "x"
READ

answer = "y"
@ 24,2 SAY "Do you wish to save these changes (Y or N)?"
@ 24,55 GET answer PICTURE "!"
READ
IF answer = "y"
  REPLACE U_TYPE WITH xtype
  REPLACE U_DESCRIP WITH xdesc
  REPLACE U_ADDR WITH xaddr
  REPLACE U_PHONE WITH xphone
  REPLACE U_CITY WITH xcity
  REPLACE U_STATE WITH xstate
  REPLACE U_ZIP WITH xzip
  REPLACE U_BEDROOMS WITH xbedrooms
  REPLACE U_BEDS WITH xbeds
  REPLACE U_BATHS WITH xbaths
  REPLACE U_LOCKOFF WITH xlock
ENDIF

USE
RETURN
PRIVATE answer
CLEAR
@ 1,0 SAY "|=================================================================|
    "
@ 1,55 SAY "============================================================|
@ 2,0 SAY "| The Reservation System |
    |
@ 2,55 SAY "Unit Maintenance               |
@ 3,0 SAY "|=================================================================|
    "
@ 3,55 SAY "============================================================|
DO addscrn
answer = ""
@ 23,10 SAY "Do You wish to <E>dit an existing Unit or <A>dd a New Unit?" GET answer PICTURE "!
READ
IF answer = "E"
   DO edunit
ELSE
   IF answer = "A"
      DO addunit
ENDIF
ENDIF
RETURN
* Program...: Display Add Unit Screen (ADDSCRN)
* Author....: Warren M. Bartlett
* Date......: 07/27/84
* Notice....: Copyright 1984, All Rights Reserved
* Notes.....:
* Reserved.: NONE
* Calls.....: NONE
* Called by: unitadd

* PARAMETERS None

@ 4,0 SAY "|=================================================================================
@ 4,55 SAY "|"  
@ 5,0 SAY "| Unit Number:"  
@ 5,78 SAY "|
@ 6,0 SAY "| Unit Type:"  
@ 6,78 SAY "|
@ 7,0 SAY "|=================================================================================
@ 7,55 SAY "|"  
@ 8,0 SAY "| Description"  
@ 8,78 SAY "|
@ 9,0 SAY "| Address:"  
@ 9,48 SAY "Phone:"  
@ 9,78 SAY "|
@ 10,0 SAY "| City:"  
@ 10,39 SAY "State:"  
@ 10,59 SAY "Zip:"  
@ 10,78 SAY "|
@ 11,0 SAY "|=================================================================================
@ 11,55 SAY "|"  
@ 12,0 SAY "| Bedrooms:"  
@ 12,31 SAY "Beds:"  
@ 12,54 SAY "Baths:"  
@ 12,78 SAY "|
@ 13,0 SAY "| Lockoff (Y or N):"  
@ 13,78 SAY "|
@ 14,0 SAY "|=================================================================================
@ 14,55 SAY "|"

RETURN
A Selected Bibliography


Byers, Robert, Anna Carlile, Debby Moody, Ellen Pader, Tom Rettig, dBASE III User Manual, Ashton Tate, Culver City, California, 1984.


