1983

A Resource system package for reservation systems

Gary Foley

Rochester Institute of Technology
School of Computer Science and Technology

A Resource System Package for Reservation Systems

by
Gary L. Foley

A thesis, submitted to
The Faculty of the School of Computer Science and Technology,
in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science

Approved by:

Warren R. Carithers
Professor Warren Carithers

Michael Lutz
Professor Michael Lutz

Peter Lutz
Professor Peter Lutz

Acknowledgements

I would like to thank my thesis advisor, Mr. Warren Carithers, for his time and guidance throughout the project. It was through his help that I was able to complete the project successfully. I would also like to thank Mr. Peter Lutz and Mr. Michael Lutz for their time spent as members of my committee. Lastly, a special thanks goes to my wife Penny for her patience and support for this project and all my studies at RIT.

ABSTRACT

The Resource System Package is a software tool developed to aid in the design and use of reservation system databases. The package is designed to automate the creation of a reservation database for arbitrary objects and to facilitate ease of maintenance and updating of existing databases. Databases set up via the resource system are maintained through the Mistress Relational Database System, although no specific knowledge of Mistress is required of either the database designer or the users of the database. The package is menu-driven for ease of use.

This package consists of two separate programs, a Resource System program and a Reservation System program. The Resource System program allows a database designer to create the reservation database. A designer is prompted for the characteristics of the objects which will be represented in the database, for descriptions of the objects available, and for information which will be used in the maintenance of the database; the program goes on to construct the actual Mistress database from the information supplied. The Reservation System program allows users to manipulate any of the available reservation databases which they have permission to modify. An individual user is presented with a set of options for manipulation of the database, which include the creation, modification, and deletion of reservations, as well as options to allow examination of the database.

This paper discusses the prepartion and development of the Resource System Package. Also discussed are the history, implementation, project description and design, and conclusions drawn.

TABLE OF CONTENTS

List of Figures

1. Introduction

Reservation Systems are basically a new concept. Computer programs which arrange to have something held for one's use at a future time or place began appearing in the early 1960's. Although they were designed primarily to eliminate manual record keeping, the implementation of such systems have resulted in saving companies much time and money. Use of a computerized reservation system allows information on the resources being reserved and information on all customers to be readily available to any employee utilizing the system. This results in less chance of errors and overbookings. The system also allows for more customers to be served at a faster rate thereby creating a larger volume per day. Of course, fewer employees are needed as a result of implementing the reservation system; therefore, the company decreases its overhead. Inadvertently, the combination of speed, accuracy, less overhead, a larger customer volume, and a better organized retrieval system, creates a more effective company.

Today's growth and progress in this field today were spurred on by the development of the SABRE reservation system which was designed to solve problems in airline passenger sales, seat inventory, and maintenance and retrieval of passenger records. This system was a joint venture between IBM and American Airlines. Fifty-six months and 400 man-years after the two companies signed a contract in October 1959, the SABRE system was installed on two IBM 7090 computers at American's computer center in Briarcliff Manor, New York. Although the project cost thirty million dollars, staff savings alone resulted in American Airlines having the competive edge over the other airlines for five to seven years.

SABRE was the first commercial, real-time reservation system; however, much of the success of SABRE should be credited to its forerunner, the U.S. Air Force's SAGE System (Semi-Automatic Guard Environment). This system, which was designed to protect the U.S. against a surprise air attack, took seven years and 1,800 man-years to develop at an ultimate cost of 1.6 billion dollars. SABRE not only was able to draw from the lessons learned through the development of SAGE, but it also utilized some of the hardware and IBM personnel assigned to the SAGE project.

Presently, there are many reservation systems available for purchase. Basically the two categories of major service are hotel and travel reservations. Several hotel packages and their respective developers are as follows:

Hotel - 2000, Unique Information Systems

Cheers, Unique Information Systems, 1981

Confirm, Prism Computer Systems Inc., 1982

Champs, Marcol Computer System Inc., 1980

Innkeepers Management System, Knox Data, 1978

Hotelplan, Diskwise Ltd., 1980

Touracc, Computeracc Service Bureau, 1981

Hotel Management, Computer Consultants of Alaska

Datahost, Cara Consulting Ltd., 1975

Hotel management, Arinco Computer Systems Inc., 1980

Hips, Point 4 Data Corp.

Wang - Based Innkeepers System, Information Management Technologies, 1977.

These packages offer management a wide range of services such as reservation tracking, billing, confirmations, deposits, arrival and departure lists, checkin/checkout, occupancy statistics, payroll, accounting, sales, cashiering, group reservations, guest messages, wakeup service, late charges, house credit limit, confirmation letters, and cancellation and no show reporting. Available travel packages and their respective developers are:

PDS Tour Operators Systems, Professional Data Systems, 1980
Vehicle Reservation Systems, O.B.I.S., 1982
Computer Systems for Travel Agents, IBM Corp., 1974
Wholesale Travel System, Computron Systems Inc..

Their capabilities consist of route bookings, reminder letters, charts, labels, reservations, origination/destination, time changes, deposits, automatic invoices, accounts, stock, customer information, and overbookings. It should also be noted that there are many in-house reservation systems presently utilized by many companies which are not available on the market for purchase by the general public .

Most of the reservation systems mentioned above are designed for specific applications and, therefore, have specialized functions such as the property management system called Hotel-2000 designed and distributed by Unique Information System Inc.. This system specializes in all aspects of hotel management. It was designed by a team of hotel managers and computer experts to provide a powerful easy-to-use management tool for small to medium size hotels, motels, and resort facilities. In addition to reservation tracking, Hotel-2000 provides for all front-desk activities, housekeeping

requirements and reports, guest accounting, and a back office function consisting of such capabilities as payroll, ledger, inventory, and mailing lists. The package is said to be "user-friendly", guiding you with menus and messages, and requires no previous computer experience whatsoever.

It can be seen that when dealing with one specific type of reservations such as hotel reservations, that many extensions can be made to the reservation function making the system very sophisticated. When developing the idea of the Resource System for this project, it was decided not to have minute, specified capabilities in the tracking of reservations because a more flexible system of reservations was desired. One system found in the research for this project was designed for generalized use rather than for a specific purpose; this was the Vehicle Reservation System developed by O.B.I.S.. This system, which was originally designed for reservations of mobile homes, was extended to include reservations of any object. The reservation system is menu driven, such that the selection of an option in the menu causes the execution of a separate program. Some of the options available are to enter dates or objects, to update the database, and to make a reservation. The system is written in the Basic programming language for use on mini computers utilizing an inhouse developed database called BIS. The system provides error checking and preformatted screens for easy interfacing. One drawback of this system is that although it can be used for any object, only one type of object can be used at a time. Also, there is little or no security checking built into the system, allowing anyone to make changes to the database at his/her leisure.

Taking the idea of the Vehicle Reservation System a step further from reservations of only one type of object at any given time, the proposed Resource System Package would allow multiple reservation systems of many different types of objects to exist at any given time. With this idea in mind, the goals of the project were outlined.

1.1 Goals

The main goal of this project, as mentioned before, was to develop a generalized resource system (as opposed to a specific resource system). A generalized system would allow reservations of many different types of objects whereas a specific resource system would be designed solely for a limited purpose. Another goal was that the programs be "user friendly". The system must be able to support multiple concurrent users and provide immediate feedback about the success or failure of a reservation attempt. It was also considered desirable that the system allow a user to quickly search a reservation database in order to decide upon and enter reservations. It was very important that the resource system insure the protection and integrity of each reservation database. Another goal was for the system to provide an online help program for learning the system and for a quick reference when a question arises. The ultimate goal of the project was to design a system that could maintain up-to-date reservation information.

1.2 Outline of the Paper

The remainder of this paper discusses the process undertaken to meet the goals set forth. In section 2, the numerous design decisions that had to be

made are discussed. Section 3 contains a detailed description of the project and its integrated parts. The fourth section is composed of many charts and diagrams depicting the various modules of the project. Within section 5 are the concluding statements followed by the Bibliography in section 6 and the Appendices in section 7.

2. Design Decisions

From the research, I was able to gain some background into the different capabilities of reservation systems available today. I wasn't, however, able to find any indepth documentation on the internal design and programming used to develop reservation systems. This project, since it was to be a "generalized" resource system, did not allow room for sophisticated capabilities such as those contained in reservation systems designed for specific types of reservations. Below are some of the design decisions, reasons for my design choices, and plans for their implementation.

2.1 Hardware/Software Tools

The main computer system within the Computer Science Department of RIT is a Digital Equipment Corporation VAX-11/780. The project was to be developed utilizing this system and the UNIX* operating system.

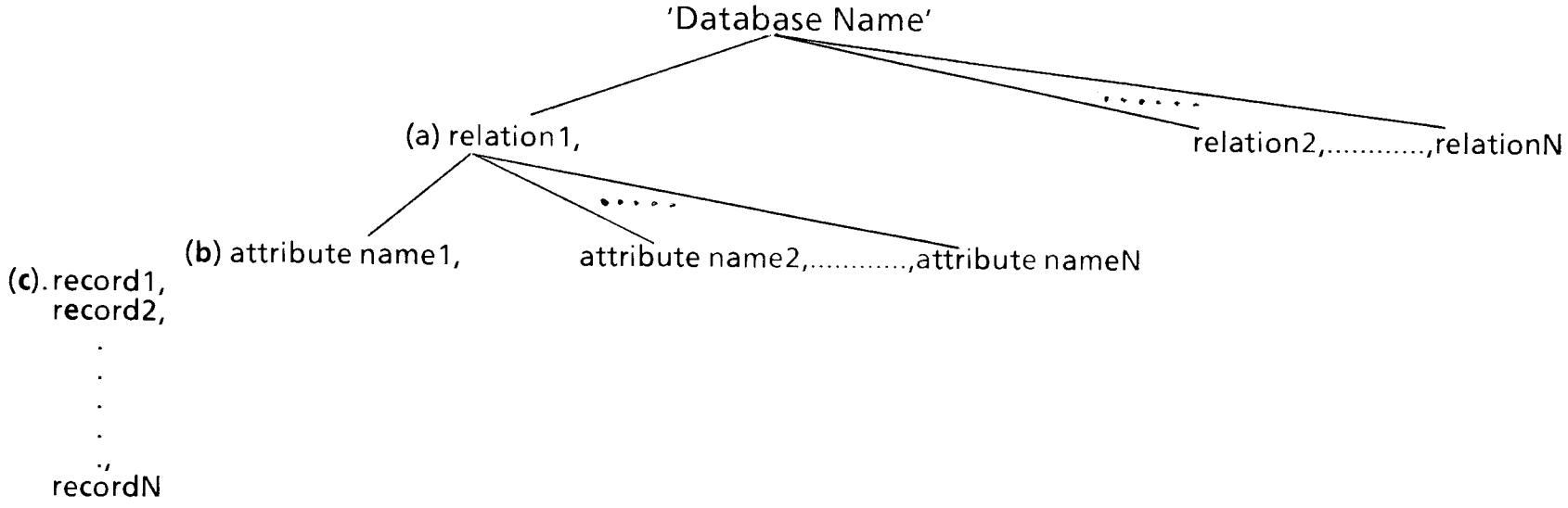
The Mistress** Database Management System was chosen as the storage medium for reservation information because of its design. Mistress is a relational database management system. Relational databases keep information in relations or tables (Figure 1). This design allows information to be easily located and updated. A hierchical database maintains

---

*UNIX is a trademark of Bell Laboratories.

**Rhodnius Incorporated, 1981.

MISTRESS DATABASE ORGANIZATION



Mistress is a relational database management system which can contain from 1 to N databases where each contains from 1 to N relations or tables(a). A relation contains from 1 to N columns which represent the attributes or characteristics of that relation (**b**). A relation also contains from 1 to N rows which represent the records in a relation (**c**).

Figure 1

information in a tree-structured fashion: for this project, such a design would require traversing the entire tree each time a value was needed. For the purposes of this project, a relational database is faster and cleaner because it allows one to go directly to the desired table for the value requested.

From the many programming languages available on the UNIX system, the C language was chosen because of its ability to interface with Mistress. This interface is provided by the Mistress Database Manipulation Procedures within Mistress. These procedures allow C programs to easily access Mistress database structures.

2.2 Design Changes

When I originally began to develop the Resource System, I had planned to allow a designer to develop his/her own menu screens and a tutorial program for each reservation system. This was to allow the designer the flexibility of how he/she wanted his/her reservation system to work. This concept, however, became impractical because of its complexity and the lack of consistency that would exist from one reservation system to another. Also, the original idea would require a large amount of storage. Therefore, I adopted the present design which consists of two programs, a Resource System program which allows a designer to develop a specific reservation system, and a Reservation System program which allows a user to utilize an established reservation system. This design solved the storage problem and provided an easier transition from one reservation system to another. There is only one tutorial for all reservation systems, and all menus are preformatted to allow for an easy step-through package.

Another design change made was not to use the Curses Cursor Movement Optimization functions. These functions are designed to minimize movement of the cursor around the terminal screen as screens are being displayed. Since it had been decided to use preformatted screens and not to have a designer develop screens, these functions were not necessary. In order to develop the preformatted screens, an easier method, the termcap database functions, was used. This is a database describing a terminal's set of capabilities. The main use of these functions was to clear the screen each time a new menu was to appear.

2.3 Implementation Approach

The Resource System program was designed and written first. This provided the means to gain familiarity with screen formatting through the use of the termcap database functions. Also, this seemed like the logical beginning, since reservations couldn't be made until a reservation system had been set up. Upon completion of the design of the Resource System program I began the development of the Reservation System program. This was designed to allow reservations to be made and updated within a reservation database developed through the use of the Resource System program. An online tutorial help function was written for each program to assist users in understanding how to use the programs. The programs were tested throughly for problems and possible infiltration of bad data into the reservation databases. The thesis paper was then written to document the project.

3. Project Description

The project, as mentioned earlier, is broken into two programs, the Resource System program and the Reservation System program. In the two sections following are descriptions of the functions of each of these programs, examples of preformatted screens, and some examples of program use.

3.1 Resource System Program

This program is written for use by a designer to develop a reservation system. The program allows a designer to set up a reservation system for any type of object desired. Many different reservation systems can be developed. Also, numerous designers can develop reservation systems at the same time. The program also provides an online tutorial for learning how to use the program and for quick referencing. Lastly, the program allows a designer to make updates to his/her existing reservation system. In the next two sections, a description is given of the program's designer interface and the program's system interface.

3.1.1 Designer Interface

The designer interface is a description of the interaction between a designer and the Resource System Program. A designer is defined as a member of the RIT faculty and staff or any other person that the Resource System Administrator (RSA, the person in charge of maintaining the Resource System Package), lists within the RSlimit file. This file contains the login ID's of those that are allowed the use of the Resource System Program. Below is an example of the development of a reservation system through the use of

the program. Upon entering the command "RS <CR>", the program gives the designer the first screen which contains the main menu of options available within the Resource System (Figure 2).

The first option is the TUTORIAL. This option allows a designer to read how to use the Resource System Program online. It also can be used as a quick reference guide for any question areas. The tutorial allows you to continue reading each major topic covered or to exit at any time and return to the main option menu. See Figure 3 for an example of a tutorial screen.

RIT RESOURCE SYSTEM
Graduate Thesis Project by Gary L. Foley

Select one option below by entering the number of your selection in the space provided.

1 TUTORIAL - how to use the Resource System.
2 SETUP - develop a reservation system.
3 UPDATE - make changes to your reservation system.
4.EXIT
Selection-->

Figure 2 - Main menu of options available within the Resource System program.

TUTORIAL - SETUP

The setup option takes you through a step by step process of designing a reservation system database. You will be prompted for various information to build the reservation database containing the objects to be reserved, the dates that reservations are accepted, and the times that reservations can be made. Before entering this option, read the remainder of this tutorial and then plan your reservation system on paper before using setup to make a reservation database system.
Enter your selection below.

1 Return to table of contents.
2 Go on to next section.
Selection-->

Figure 3 - Example of a tutorial screen within the Resource System Program.

The second option, the SETUP option, is the main selection in the Resource System program. Here the designer is told step by step how to create a reservation system. The first screen to appear allows the designer to name his/her reservation system (Figure 4). Once the name is selected, the second screen prompts the designer for a list of user and group ID's to help restrict the use of this particular reservation system being developed. A designer can list ID's of both those who are and those who are not allowed access to his/her reservation system. A file is created which stores these values. The next screen has the designer enter the characteristics or attributes of the object(s) to be reserved. In actuality, the designer is creating the attributes for the object table. Upon completion of the entry of all characteristics, the designer is prompted for values of each of the characteristics which make up the records of the table.

SETUP

The reservation systems presently available are:

roomres
terminal
books

Do not use any of the above names for your reservation system. The database name is limited to 12 characters in length.
What is the name of your reservation system?--> AVequipment
Database name: AVequipment
Is this correct? (Y or N)--> Y

Figure 4 - First screen in SETUP option used to name the reservation database being built. (Example given to build a reservation database.)

Once the object table has been built, the designer is prompted for the key characteristic (one previously entered in the set of characteristics) that will uniquely identify the object being reserved. This key characteristic will be one of the attributes in the table which will hold actual reservations; the other attributes in this table are the date and time of the reservation, and the ID of the person making the reservation.

The next table that the designer will help build is the date table. Here the designer must enter each date upon which reservations will be available. The date must be in the format MM/DD/YY where MM is the month, DD is the day, and YY is the year. (Example: October 21, 1983 would be entered as 10/21/83.) The system checks for valid dates and will only allow reservations for the present year up to 1989. It is suggested that a calendar be used when entering the dates.

The last table to be built is the time table. The designer is first prompted for a time interval which represents the time span allowed for each reservation. This interval can be anything from thirty minutes to two and one-half hours. The designer is next prompted for the starting time that reservations will be accepted and the ending time of the last reservation each day. All times must be listed according to military standards. An example of a reservation database and the four tables is given in Figure 5.

EXAMPLE OF A RESERVATION DATABASE

Room Reservations

a) Object

bldgnum-roomnum	*capacity*
12-1000	30
12-1001	45
12-1002	105
12-1003	50
12-1004	60
12-1005	60
12-1006	45
12-1007	30
12-1008	75
12-1009	25
12-1010	50

b) Dates

date-available
09/07/83
09/08/83
09/09/83
09/12/83
09/13/83
09/14/83
09/15/83
09/16/83
09/19/83
09/20/83
09/21/83
09/22/83
09/23/83
09/26/83
09/27/83
09/28/83
09/29/83
09/30/83

c) Time

slots
0800
0830
0900
0930
1000
1030
1100
1130
1200
1230
1300
1330
1400
1430
1500

d) Reservations

bldgnum-roomnum	*date*	*time*	*who-reserved*
12-1005	09/14/83	0900	spn2249
12-1002	09/12/83	1330	tjt8931
12-1002	09/12/83	1400	tjt8931
12-1010	09/08/83	0800	wdj3374
12-1010	09/08/83	0830	wdj3374
12-1010	09/08/83	0900	wdj3374
12-1007	09/16/83	1400	ptr7204
12-1000	09/20/83	1200	spn2249
12-1000	09/21/83	1200	spn2249
12-1002	09/19/83	1330	tjt8931
12-1002	09/19/83	1400	tjt8931
12-1004	09/15/83	1500	dkd3624
12-1010	09/30/83	1300	dkd3624
12-1000	09/16/83	1300	kcl5399
12-1010	09/08/83	1000	pjf3240
12-1010	09/30/83	1330	dkd3624

Figure 5 - Example of a reservation database. a) - the object table containing the objects available to be reserved and their characteristics. b) - the date table containing the dates that reservations are available. c) - the time table of time slots when reservations are acceptable. d) - the reservation table listing the existing reservations presently within the database.

Now that all the input necessary to build the tables has been given, the designer is asked for a daily limit to the number of reservations. A daily limit of zero is taken to mean that reservations are unlimited. Any other number entered would limit a user to making no more than that number of reservations per day.

The designer is also given two options to determine the frequency of reservation cleanup. A message will be sent to the RSA either each week or each month depending upon the selection by the designer. The RSA would then relay that message to the designer or do the cleanup himself/herself.

It is at this point in the Setup option that the database is created. The designer is informed that a series of *'s will appear accross the screen as the database is being built. Each * represents a Mistress command being executed. When all information has been inserted into the database, the final menu will appear.

The final menu within the setup option allows the designer to display the tables built to check for accuracy (Figure 6). After the displays are complete, the designer is brought back to the main menu of choices (Figure 2).

You may verify the tables built by making a selection below.

1 Display the object table.
2 Display the date table.
3 Display the time table.
4 Display the reserve table.
5 Display all tables.
6 End of requests.
Selection-->

Figure 6 - Final screen in the SETUP option allowing the designer to verify that all the tables created contain the correct information.

The third and last option from the main menu is the UPDATE option. This option is used to allow a designer to make changes to his/her existing reservation database. Within this option the designer must first give the name of the reservation database that is to be updated. He/she is then given the choice to delete some values from a table, to add some values to a table, or to change the user or group ID's within the user limit file for that reservation database. The designer is prompted for the necessary input to make the needed changes. For example, if the designer was to cleanup old data from the date table, all existing reservations on those dates would also be deleted. The update function must be used very carefully to maintain the integrity of the database. When the designer leaves the update function, the main menu appears allowing the designer to read the tutorial, setup another reservation system, update an existing reservation system, or to exit the Resource System program.

3.1.2 System Interface

The system interface describes the activities that occur within the Resource System Program that do not require any interaction with a designer. These activities occur automatically when a designer is utilizing the program. The following paragraphs describe these activities.

When a designer enters the Resource System Program, the program automatically checks for authorization to use the program. In order for a designer to use the Resource System Program, his/her login ID must be

contained in the RSlimit file of authorized users. If the ID is not within this file, the designer is given a message and is terminated from the system.

Once within the Resource System Program, the designer is automatically prompted for input through preformatted screens (described earlier). Each time input is accepted, the next screen appears guiding the designer through the program until he/she chooses to leave the program.

Within the setup option, the designer's login ID is placed into the specification directory in a file. This file is checked for authorization each time that the reservation database is to be updated. Only the designer of the reservation database or the RSA is allowed to update the reservation database. When a designer specifies the key characteristic of the object(s) to be reserved, this key characteristic is stored within a file k."database name" in the specification directory.

Throughout the program, the designer's input is checked for validity before being accepted. When setting up user limits (who is or is not allowed access to your reservation system), there are two fields that are checked automatically. The first is the limit designator. This specifies whether the limit being entered is for a user ID or a group ID. If a user ID is going to be entered, the limit designator must be a capital 'U'. If a group ID is going to be entered, the limit designator must be a capital 'G'. Any other entry will flag an error message and the value must be re-entered. The second value checked is the usage limiter which tells whether or not the user ID or group ID entered is or is not allowed usage of your reservation system. This is determined by your entering a ' + ' for usage allowed or a '-' for no usage

allowed. An error message is received for any values entered other than ' + ' or '-'. During the screen for entering characteristics, if a duplicate characteristic is entered, it is flagged. Dates entered are checked for correct format (MM/DD/YY), valid months, valid days, valid years, and duplicates within the date table. Times entered are checked for valid times, time ending in 00 or 30, starting time less than ending time, and valid time intervals. In screens where a menu of numbered options is given, the program checks for a valid selection, giving an error message if an illegal choice is made.

3.2 Reservation System Program

This program is the only one most users of reservation databases will ever see. The program allows a user to enter and update reservations in a selected reservation database. Multiple users can utilize the program simultaneously. An online tutorial is available to assist new users in learning how to use the Reservation System program and for use as a quick reference. The program is a "user-friendly" system that provides preformatted screens and menus to help guide the user through a reservation session. It also checks for valid input to ensure that errors do not creep into the database. In the next two sections, a description is given of the program's user interface and the program's system interface.

3.2.1 User Interface

The user interface is a description of the interaction between a user and the Reservation System program. A user is defined as any user login ID or member of a group login ID contained in the limit file developed by the

designer of a particular reservation database. Following is a description of the use of the program from the user's perspective. Some diagrams and examples to further explain its use are also given.

By entering the command "ressys <CR>", the user is taken into the Reservation System program and shown the first screen. Here, the user is shown the name of all existing reservation databases and asked to select the one in which he/she would like to work. After a reservation database has been selected, the second screen appears displaying the eight main choices in the Reservation System program (Figure 7).

The following is a list of choices to search the _____ reservation system in order to determine your reservation. Enter your selection below. When you're finished viewing the table, enter <CR>.

1 TUTORIAL - how to use the Reservation System program.
2 List time table containing time slots of when reservations are accepted.
3 List date table containing dates of when reservations are accepted.
4 List object table containing the characteristics of the objects available to reserve.
5 List existing reservations according to a particular attribute.
6 Make a reservation.
7 Update a reservation - to make changes and deletions.
8 End of reserve session.

Selection-->

Figure 7 - Main menu of choices within the Reservation System program.

The first choice available is to read the TUTORIAL. The tutorial is a group of screens designed to aid a user in learning how to make use of the Reservation System program or to be used as a quick reference. The tutorial can be read in its entirety or by choosing particular topics.

The second, third, and fourth choices are similar in that they allow a user to view a database table. The second selection will display the time table containing the times that reservations are available (Figure 5c). After selecting the third option, a user would be shown the date table of dates that reservations are available (Figure 5b) If option four is chosen, the object table of the objects available to be reserved is shown, along with each object's characteristics (Figure 5a). These choices help a user to plan his/her reservation.

In choice five, the user is given a variety of choices as to how he/she would like to view or search the existing reservations. There are eight choices within this screen (Figure 8). The first allows a user to view all existing reservations on a particular date (Figure 9a). The user would enter the date in question. The second option lets the user view all existing reservations of a particular object (Figure 9b). Within option three, the user would be prompted for a person ID to view all existing reservations made by a particular person (Figure 9c). The fourth option requires the user to enter a date and an object to display all existing reservations of that object on the date given (Figure 9d). The fifth option allows a user to search the reservation database according to person ID and date (Figure 9e). The sixth option is similar but it searches by person ID and object (Figure 9f). The seventh option, possibly the most useful when searching for a reservation slot, displays for the user all objects not reserved within a particular time span on a specific date (Figure 9g). This would be helpful when the user needs to make a reservation on a particular date and has only certain times free in his/her schedule. The eighth and last option simply is to exit the reservation search option and return to the main menu or first screen of options to continue.

Each reservation includes the following:
object reserved date reserved time reserved person ID

Make a selection below as to how you would like to search the data.

1 *List all existing reservations on a particular date.*

2 *List all existing reservations of a particular object.*

3 *List all existing reservations made by a particular person.*

4 *List all existing reservations on a particular date of a specific object.*

5 *List all existing reservations made by a particular person on a particular date.*

6 *List all existing reservations of a particular object made by a particular person.*

7 *List all objects NOT reserved during a particular time span on a specific date.*

8 *Exit reservation display.*

Figure 8- Screen of options to aid a user when searching for an available reservation slot.

a)

bldgnum-roomnum	date	time	who-reserved
12-1007	09/16/83	1400	ptr7204
12-1000	09/16/83	1300	kcl5399

b)

bldgnum-roomnum	date	time	who-reserved
12-1010	09/08/83	0800	wdj3374
12-1010	09/08/83	0830	wdj3374
12-1010	09/08/83	0900	wdj3374
12-1010	09/30/83	1300	dkd3624
12-1010	09/08/83	1000	pjf3240
12-1010	09/30/83	1330	dkd3624

c)

bldgnum-roomnum	date	time	who-reserved
12-1005	09/14/83	0900	spn2249
12-1000	09/20/83	1200	spn2249
12-1000	09/21/83	1200	spn2249

d)

bldgnum-roomnum	date	time	who-reserved
12-1010	09/08/83	0800	wdj3374
12-1010	09/08/83	0830	wdj3374
12-1010	09/08/83	0900	wdj3374
12-1010	09/08/83	1000	pjf3240

e)

bldgnum-roomnum	date	time	who-reserved
12-1002	09/19/83	1330	tjt8931
12-1002	09/19/83	1400	tjt8931

f)

bldgnum-roomnum	date	time	who-reserved
12-1004	09/15/83	1500	dkd3624

g)

bldgnum-roomnum
12-1001
12-1002
12-1003
12-1004
12-1005
12-1006
12-1008
12-1009
12-1010

Figure 9 - All examples above are from using the search options in figure 8 on the Room Reservation database in Figure 5. a) All reservations on 09/16/83. b) All reservations of object 12-1010. c) All reservations by person ID spn2249. d) All reservations of object 12-1010 on 09/08/83. e) All reservations made by person ID tjt8931 on 09/19/83. f) All reservations made by person ID dkd3624 of object 12-1004. g) All objects NOT reserved between 1200 and 1500 on 09/16/83.

Choice six is the key selection to the Reservation System program because it is through this choice that reservations are made. Once a user has determined the date, time, and available object for his/her reservation, this option can be selected. The user is prompted for the date, time, and object to be reserved. The user is informed of the number of reservations that are allowed per day and his/her total thus far the the day entered. As long as no errors occur in making the reservation, a message is returned that the reservation has been made successfully. (Possible errors are discussed in the section entitled "System Interface".)

Choice seven is used for updating an existing reservation. Two options are given (Figure 10). The first is to delete a reservation. Within this option, the user enters the date, time, and object of the reservation to be deleted. If the request is completed without error, a message is returned that the deletion has been accomplished. The second option allows the user to change an existing reservation. Here again the user must enter the date, time, and object of the existing reservation. Then he/she must enter the time, date, and object of the revised reservation. If the request is completed without error, a message appears that the change has been made successfully.

At the end of each of the above seven selections, the user is returned to the main menu consisting of these seven selections and selection eight which when selected, the user leaves the Reservation System program. It is not until selection eight is made that a user leaves the program.

UPDATE

The following is a list of options to choose from when making updates to your current reservation.

1 Delete a reservation.
2 Change a reservation.
3 Exit update.
Selection-->

Figure 10 - Screen to allow user to select an update option.

3.2.2 System Interface

The system interface describes the activities that occur within the Reservation System program that do not require any interaction with a user. These activities occur automatically when a user is utilizing the program. The following paragraphs contain a description of these activities.

The first screen of the Reservation System program prompts the user for the name of a reservation system that he/she would like to work with. The system automatically checks to see that the user is authorized to use the particular reservation database entered. The user limit file for the database in question is checked for the login ID of the user. If it is not contained within this file or is contained but has a no usage indicator (represented by a '-'), the user receives a message and the program ends.

Any time a reservation database is entered, the program checks a date file for that particular reservation database. Within the date file is the date and time of the last cleanup message. Depending upon the option chosen by the designer at setup time, the program will determine whether a cleanup message is due to be sent. Only if the time period specified (weekly or

monthly) has passed, will another message be sent. The date file is then updated with the present date and time.

When making reservations, the reserve table must be locked in order to make an update. The program checks to see if the database is open so that a user can enter his/her reservation. If it is locked and in use by someone, the program continually sleeps for a short time and then tries again to enter the database. This continues for sixty seconds before it asks the user if he/she wants to try again or try later. Once a user has the reserve table locked to allow the entering of his/her reservation, the program checks the reservation limit file for the maximum number of reservations allowed per day. A comparison is made between this number and the number of reservations presently made by this user on the date that the user has entered for his/her reservation.

The program checks to see that all dates, times, and objects entered are valid and exist within the database tables. If a value is invalid, the user receives a message and is told to try again. If a value does not exist within the database tables, the user receives a message and is returned to the main menu. If the user's reservation is valid, but an entry already exists for that date, time, and object, a message is given and the user is returned to the main menu. If all goes well, the user receives the message that the reservation has been made successfully.

When a user decides to update an existing reservation, the reserve table must be available so that it can be locked from other users until the change has been made. The program checks this and sends a message to the user if it

cannot lock the table. Once the user has the reserve table locked, he/she can continue making the change. The program checks to make sure that the record entered to be updated belongs to the user making the request. A user can only update his/her own reservations; if one tries to update a reservation which does not belong to him/her, a message is printed that the update cannot be made and the user is returned to the main menu. If the change can be made, the user receives a message that the update occured successfully.

The program provides for the possible problem of contention between multiple users utilizing the program at the same time. During requests to read the database, each user's request is handled as a separate set of mistress commands which will execute in a first-in-first-out manner. Each request is built in a file unique to the user making the request. The file name is made up of the process ID (system ID given to each request) and the letters "prog". This enables each user request to be unique. During requests to update a reservation database, as mentioned before, the program will sleep between attempts to gain exclusive access to the database table. This command allows the user to try a specified number of times rather than to try once and possibly not gain access. There will be times when a user may have to wait for the database table to be freed.

4. Module Designs

The Resource System can be broken down into a variety of ways to explain its structure. The following section discusses the breakdown used for this project. The file structure is an explanation of how the program's functions and files are stored. The call structure is the calling sequence of the various functions that make up each program. The control flow is the sequence that the program follows to perform a user/designer request. The data flow is the description of how information is processed within each program. Below is a more indepth explanation of each of these structures supplemented by annotated diagrams.

4.1 File Structures

There are fifty or more different functions that make up the Resource System Package (figure 11). The functions are stored within the Func directory. Also within the Func directory is a sub directory called the dbase directory. The dbase directory holds the Mistress databases of which each is a directory containing the database files. The dbase directory also contains a subdirectory called the spec directory, which contains the specification files for each database. Within these files is information such as the designer of the database, the date and time of the last cleanup message, and the maximum number of reservations allowed each day. Figure 12 shows a diagram of the file structure.

4.2 Call Structure

The calling structure is the possible sequence in which the functions that make up the Resource System Package can be called. Both the Resource System program and the Reservation System program utilize some of the same functions, although the majority of the functions are used in either one program or the other.

Upon entering the Resource System program, three choices are given (figure 13). Each of these choices and its call structure is shown. The setup option does not give choices but steps a designer through each function simultaneously (figure 14). The update option does allow choices and any of the branches within the diagram of figure 15 can be taken. The tutorial option also allows choices so that one or all of the functions can be called.

Within the Reservation System program, a user is given a main menu of options. Any of the functions may be called depending upon the user's request. Some of the main menu options have sub menus with further choices within that option. Figure 16 shows a diagram of the call structure of the Reservation System Program.

4.3 Control Flow

The flow of control is how the program allows a designer or user to step through the program. Each program (Resource System or Reservation System) provides several initial choices, each of which has a specific sequence of steps to accomplish the request. Figures 17 and 18 display the control flow

for the Resource System program and the Reservation System Program respectively.

4.4 Data Flow

Data flow is the flow of information. The main tools necessary to enable the flow of data are the designer or user, a CRT terminal, a disk storage device, a CPU, and lines connecting them all. Figure 19 shows the data flow for the Resource System and the Reservation System Programs.

a)
adddates
addobject
addtime
attribute
bldddate
bldrelation
buildlist
check
cleanup
datetable
disdate
disobj
display
disres
distime
entval
getatt
getname
getnum
main
reserve
reslim
sendmail
setlim
setup
strsave
timecheck
timetable
tutadd
tutchar
tutchnglim
tutcleanup
tutdate
tutdelete
tutdisplay
tutorial
tutreserve
tutreslim
tutresname
tutsetup
tuttime
tutupdate
tutuserlim
upadd
upclean
update
updelall
updelatt
updisplay
values

b)
check
getatt
getname
getnum2
mkreserve
nresdatetime
opensesame
resdate
resobj
resobjdate
resobjper
resperdate
resperson
ressys
resupchng
resupdate
resupdel
rtutorial
strsave
timecheck
tutrdate
tutrmake
tutrobj
tutrselect
tutrsearch
tutrtime
tutrupdate

c)
catres
contents
dbase
global
Makefile
resout
ressys
RS
thesisfile
thesis1

Figure 11 - List of files within the func directory. a) - functions that make up the Resource System Program. b) - functions that make up the Reservation System Program. c) - Other files within the directory.

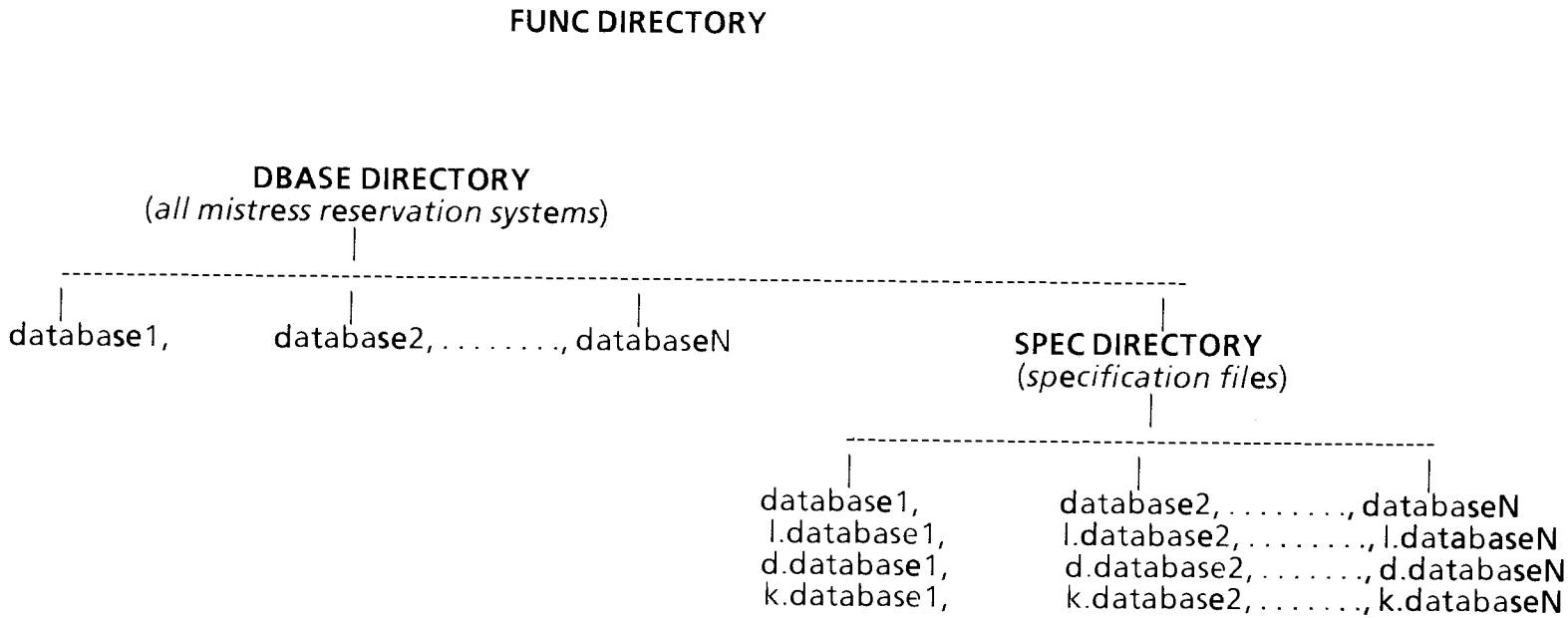


Figure 12 - File Structure

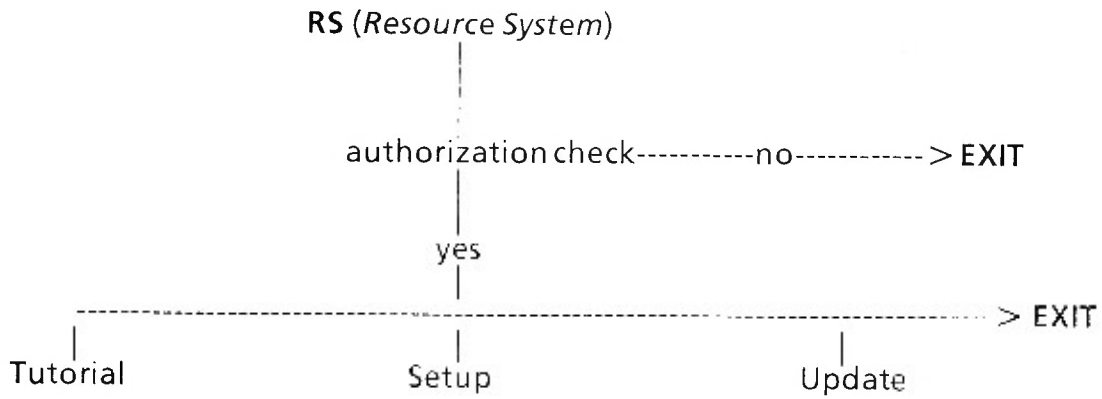


Figure 13 - Call Structure - Main menu of choices witin the Resource System Program.

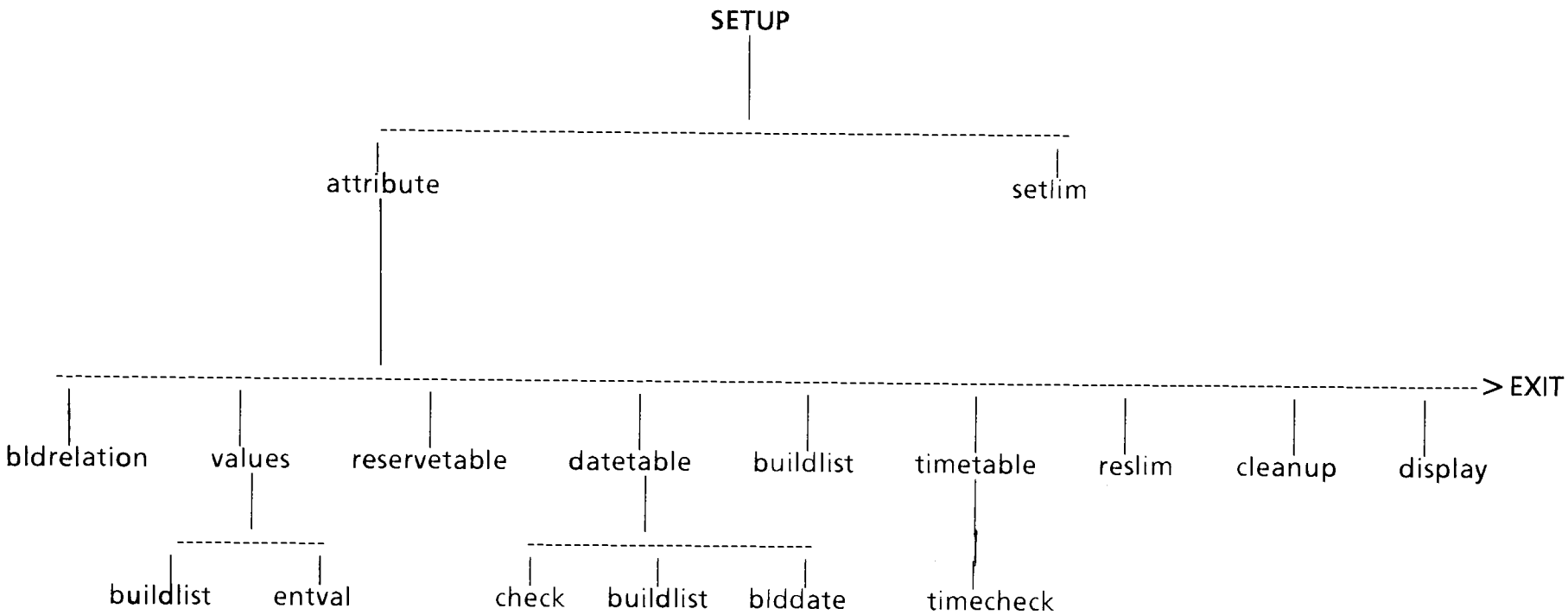


Figure 14 - Call Structure for the Setup option of the Resource System Program

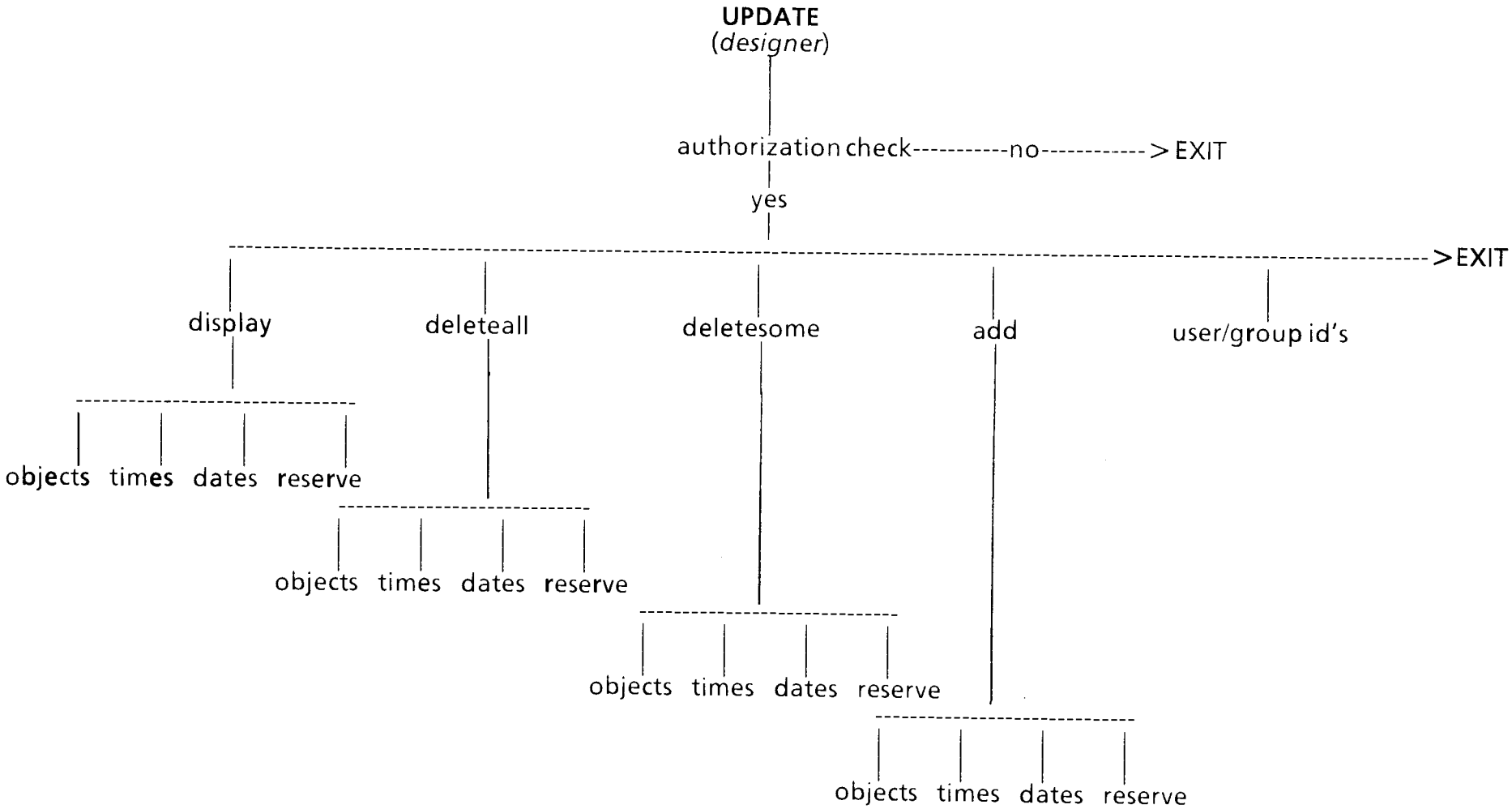


Figure 15 - Call Structure for Update option within the Resource System Program.

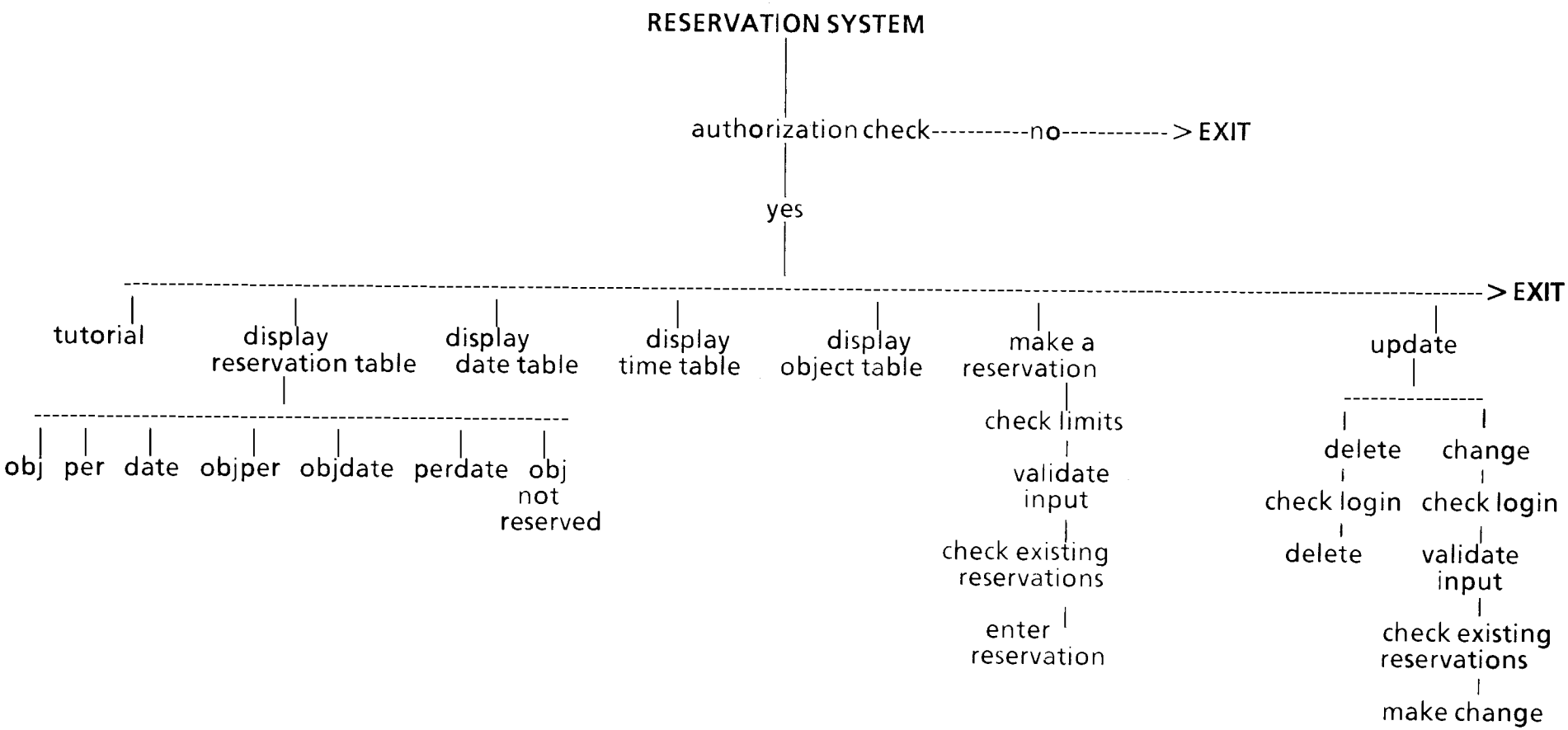


Figure 16 - Call Structure for the Reservation System Program.

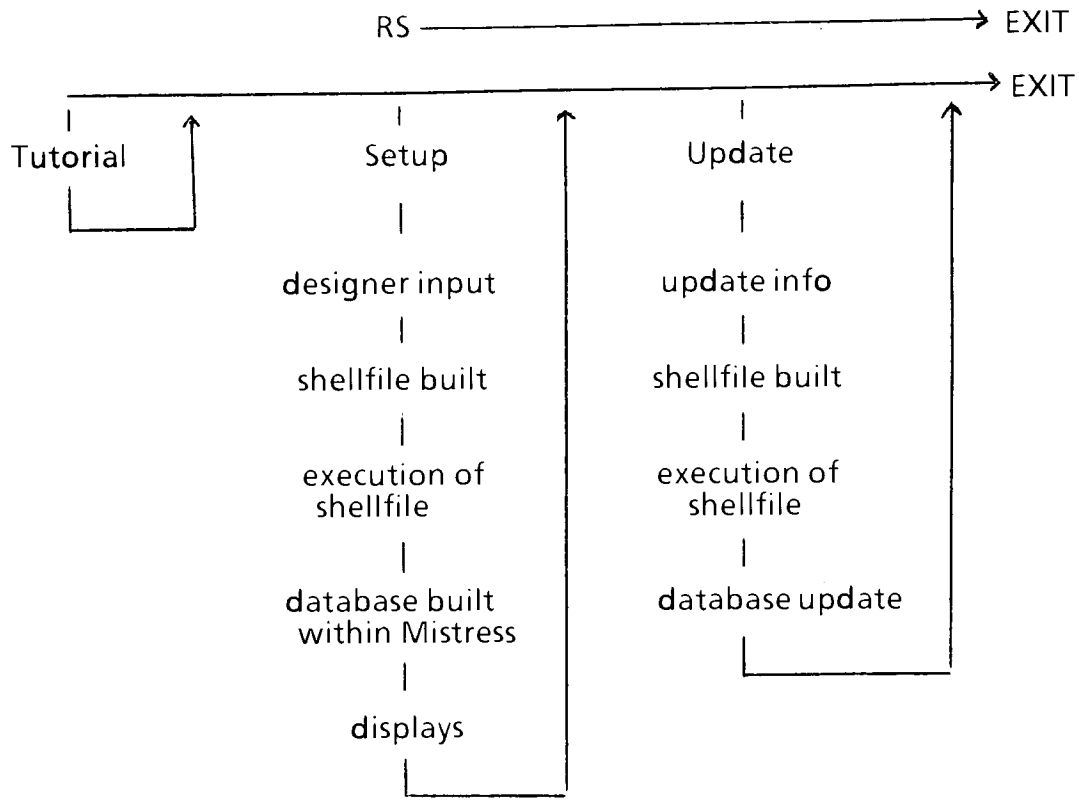


Figure 17 - Control Flow for the Resource System Program.

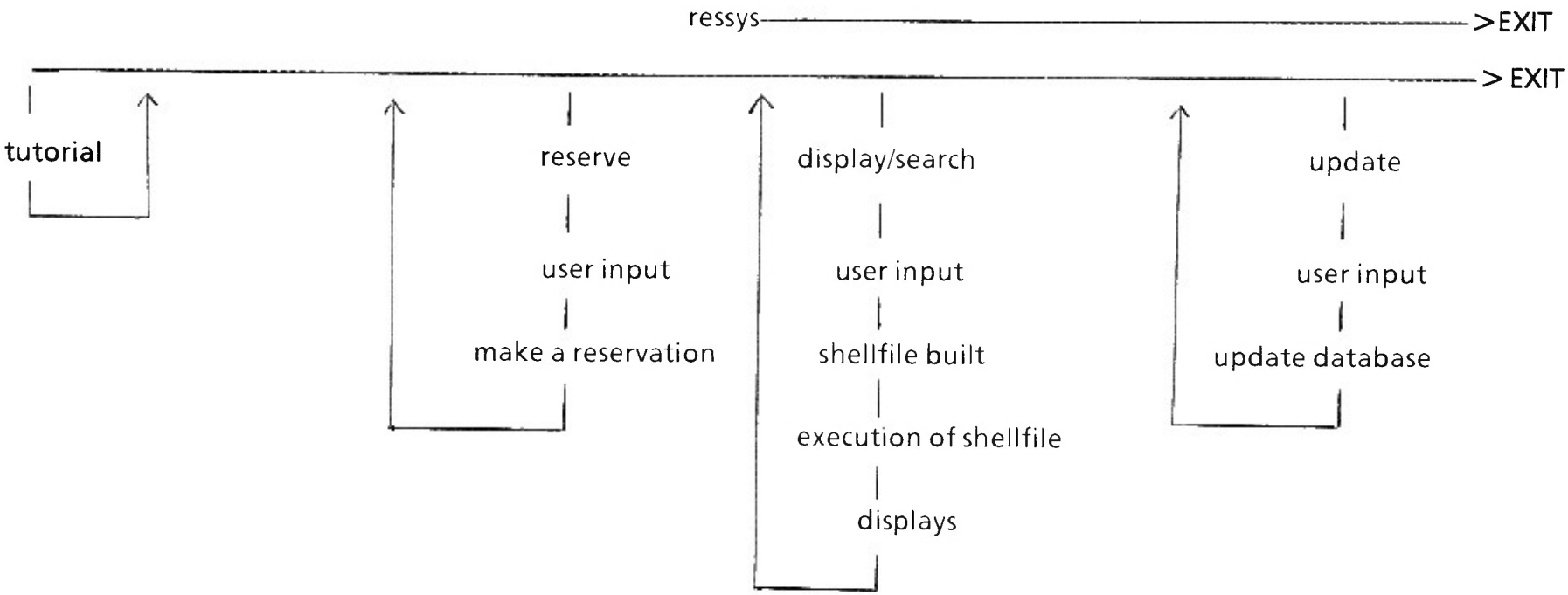


Figure 18 Control Flow for the Reservation System Program.

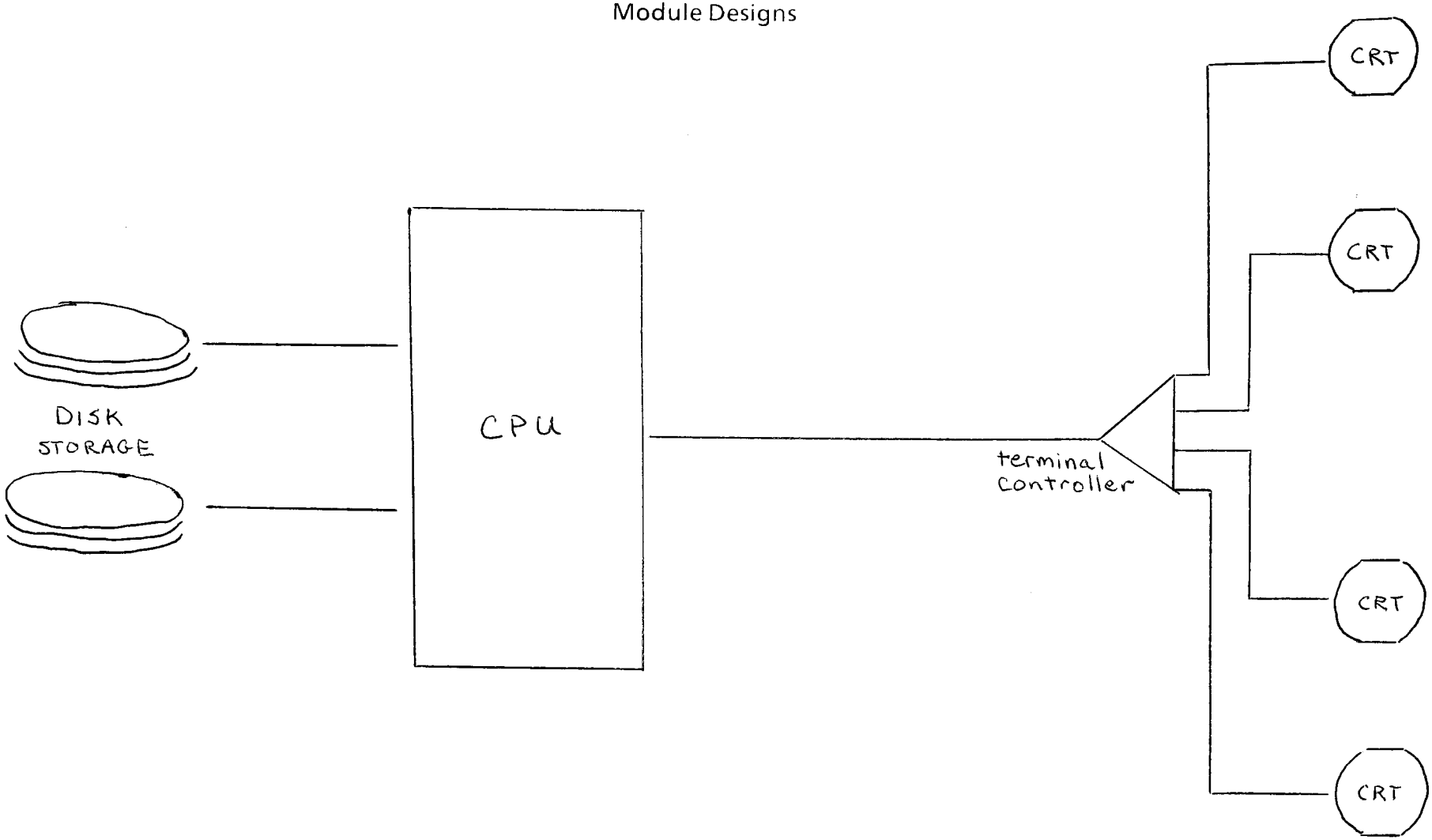


Figure 19 - Data Flow for Resource System Program and Reservation System Program.

5. Summary

This project has resulted in a "generalized" Resource System Package for the development of reservation systems and the tracking of reservations within those systems. Chapter 1 provided the historical background and goals of the project. The design decisions that had to be made along with the approach to implement them were discussed in Chapter 2. Chapter 3 was a detailed outline of how the package works along with diagrams and examples. Through various charts and diagrams in Chapter 4, the module designs of the file structure, call structure, and data flow were displayed.

5.1 Conclusions

At the start of this project, I was somewhat apprehensive about the scope of the entire project and the procedure to best accomplish the task. Having limited experience using the C programming language, the Mistress database, and the UNIX operating system, I also worried about my ability to make the project functional. Now that the project is complete, I am by no means an expert in the above mentioned areas, but definitely am more comfortable and knowledgeable about their uses. I hope that this project will be a useful tool. Below are some limitations and restrictions of the project, and some suggestions for alternative approaches and future extensions.

5.1.1 Limitations and Restrictions

There are a number of different limitations and restrictions within both the Resource System program and the Reservation System program. All

entries are checked for accuracy although some entries have no limits on them except maybe the length of the word or characters. For entries such as this, the user is shown what he/she entered and asked to verify that it is correct. If not, he/she can re-enter the item. The system limits an object to having one unique characteristic that would identify the object. It does not allow for combined characteristics to uniquely represent an object. Another limitation is that the designer can only specify a limit to the reservations made daily. It might be desired to specify weekly or monthly limits also. Reservation slots are limited to from thirty minutes to two and a half hours in thirty minute intervals. All times entered must end in 00 or 30. Time is kept according to military standards which may become confusing. The restriction on dates is that they must be in the format MM/DD/YY. The system will not catch dates out of sequence. The day of a month can be up to thirty-one days but in a month that has only thirty days, the thirty-first would be accepted. Reservations can only be made for the person making the reservation. Reservations can only be updated by the owner of that reservation. A final limitation to the package is that every day is restricted to the same time slots. Varying time slots for each day is not allowed.

5.1.2 Alternative Approaches for an Improved System

One problem that occurred frequently was that I would need the same code written in a previous function as in the one that I was creating. At times I was able to make subfunctions to be used again and again. To improve the system, however, I would suggest breaking down some of the duplicate code into separate functions. Another suggestion to improve the system might be to utilize the Database Manipulation Procedures more often instead of shellfiles to execute Mistress commands.

5.1.3 Suggestions for Future Extensions

My first suggestion for an extension would be to develop some programs to give the RSA (Resource System Administrator) some diagnostics about the use of the system. For instance, it may be useful to know how often the Resource System is being used or how often a particular reservation system is used. It might also be helpful to know who and how often certain individuals are using the system.

Another extension to the system might be to allow for multiple key characteristics. For example, if the objects to be reserved were chairs, the chair number, room number, and building number would be necessary to uniquely represent that chair.

Another possible change to the package would be to allow for a separate time table to exist for each day of reservations. This would allow a designer to specify varying time slots for different days. For example, weekend and holiday time slots may be different from the regular work week.

A final suggestion would be to allow the designer to specify a weekly limit to the number of reservations a user would be allowed to make. To do this, one would have to determine what day of the week it was that the reservation was being made and the number of reservations made previously that week.

The project was designed for the RIT faculty and staff for possible uses such as terminal reservations, conference room reservations, presentation

seating reservations, etc.. The project could be utilized in many other ways in varying environments for vehicle reservations, film reservations, high school/college scheduling, theatre reservations, restaurant reservations, rent-a-center rental, banquet hall reservations, costume rental, and doctor, dental, or beautician appointments. Regardless of the application of this project, it should provide for a faster, more accurate, and better organized system for reservation and retrieval of information.

6. Bibliography

Arnold, Kenneth C.R.C.. "Screen Updating and Cursor Movement Optimization: A Library Package", University of California, Berkeley.

Bourne, S. R.. The UNIX System. Addison-Wesley Publishing Company. 1983.

Busch, David D.. "Travel Planner". Interface Age. May 1981. pp. 102-104.

"Computerized Registration Eases Convention Line Lag, Aids Information Queries". Computerworld September 1, 1980. pp. 14-21.

Graham, Neil. Introduction to Programming. West Publishing Company. 1980.

Kernighan, Brian W. and Dennis M. Ritchie. The C Programming Language. Prentice-Hall, New Jersey. 1978.

"Mistress: The Host Language Interface". Draft Version 3. Rhodnius Incorporated. 1981.

"Mistress: The Query Language". Draft Version 2. Rhodnius Incorporated. 1981.

"Rental Agents Provide Faster Service with Terminal System". Infosystems. November, 1981. pp. 102, 104.

"Reservation Facility Links 2,500 Hotel Chain". Computerworld. May 5, 1980. pp. 14,50.

Roberts, Wayne. "Travel Business Gets Boost From Computers". Compdata. February 1980. pp. 5, 34-35.

"SABRE - Realtime Benchmark has the Winning Ticket". Data Management. September, 1981. pp. 26-28.

"Special Sales and Distribution Functions". Datapro Reports on Software Products. Datapro Research Corporation. November, 1982.

Tenenbaum, Aaron M., and Moshe J. Augenstein. Data Structures Using Pascal. Prentice-Hall Incorporated., New Jersey. 1981.

Thompson, K. , and D. M. Ritchie. UNiX Programmer's Manual. Seventh Edition. Bell Laboratories, New Jersey. 1978.

Tsichritzis, Dionysios C. and Frederick H. Lochovsky. Data Models. Prentice-Hall, New Jersey. 1982.

7. Appendix

A. Glossary of Key Words and Phrases

attribute	-	characteristic of the relation.
authorization	-	permission.
characteristic	-	value that describes an object.
cleanup	-	process of deleting all old data from a database.
contention	-	two or more users trying to lock a file that is already locked.
<CR>	-	carriage return.
current reservations	-	those presently existing within the specified database.
database	-	storage area for each reservation system.
date format	-	MM/DD/YY where M is month, D is day, and Y is year.
date records	-	all entries within the date table.
date table	-	a mistress relation to hold the dates that reservations are allowed.
designer	-	one who develops a reservation system using the Resource System.
designer level	-	developing reservation systems utilizing the Resource System Program.
display	-	show.

generalized resource system - a system designed to allow multiple reservation databases to be maintained for many different types of reservation systems.

group ID - a group identification for a related number of users.

group limit - the letter G is used when a particular group is being specified in the limit file.

interactively prompted - the program asks the user to enter information on the terminal.

key characteristic - value that solely identifies a reservable object.

limit designator - a value G to represent a group limit, or U to specify a user limit.

lock - a relation within a database must be locked exclusively to the user making the update to that relation.

military time - 2400 time as kept by the military.

Mistress - relational database used for storage.

name of the account - the login ID of those who may or may not use the reservation system.

object - resource to be reserved.

object records - all entries within the object table.

object table - a mistress relation to hold reservable objects and their characteristics.

option - list of choices.

relation - a table built in the Mistress database.

reserve records - all entries within the reserve table.

reserve table - a mistress relation to hold reservations.

reservation database - a Mistress database for the tracking of reservations.

Reservation System Program - a C interactive program used to enter and update reservations within the reservation database specified.

Resource System Administrator - person who maintains the programs within the Resource System Package and who handles all problems with the reservation databases.

Resource System Package - name of this project.

Resource System Program - a C interactive program used to design a reservation database.

ressys - command used to execute the Reservation System Program.

RS - command used to execute the Resource System Program.

RSA - see Resource System Administrator.

RSlimit file - name of storage area for the user and group ID's of those whom are allowed usage of the Resource System Program.

screen - the CRT terminal - what appears on the terminal.

span of dates - a range - ex 07/21/83 to 07/28/83.

specific resource system - a system designed only for a certain type of reservation.

table - listing of records within a relation.

tailor designed reservation system - a reservation system developed to the specifications of the designer.

time interval - the segment of time for each reservation.

time records - all entries within the time table.

time slots - an interval of time that a reservation can be made.

time table - a mistress relation to hold the times that reservations are allowed.

tutorial - online help program.

unique identifier - the characteristic that can solely identify a reservable object.

Unix command file - a list of commands stored together to be executed at a later time on the Unix/Vax computer system.

usage limiter - a + represents that usage is available and a - signifies that usage is not available.

user friendly - uses an easy to interact with format.

user ID - the login identification of a user.

user level - making and updating reservations using the Reservation System Program.

user limit - the letter U is used when a particular user is being specified in the limit file.

values - data associated with the characteristics of objects.

B. Internal Functions

The following is a list of some of the internal UNIX functions, commands, and system calls utilized within the programs and a description of how they were used.

1. access - used to check to see if a file entered exists. This is used when a user or designer selects a reservation system to work with.

2. chdir - command used to change directories. It is used to move among the three directories of the Resource System Package.

3. chmod - command used to change the mode of a file. It is used to change files to an execute mode.

4. getlimits - retrieves a value from the limits file stored within the specification directory. In this project, a ' + ' allows access to a reservation system and a '-' restricts one from a reservation system. This function is used to retrieve the limit value associated with the login ID of the user.

```
U  glf8954     +
G  root        +
G  fac         +
G  ungrad      -
G  grad        +
U  sss1982     +
U  wrc3315     -
```

Example of what the limit file for a particular reservation system might look like.

5. getlogin - returns the login ID of the person calling the function. This is used to retrieve the designers ID to be stored within the specification directory for future protection of the design of his/her reservation system. The only one that can update a particular reservation system is the designer of that system. This function is also used to retrieve the user's login ID at the time of making a reservation. By using this function, a user can enter a reservation only for himself/herself. Getlogin is also used when a user attempts to update a reservation, checking that the reservation to be updated is his/hers.

6. getpid - gets the process identification. This is used to uniquely identify each user's request within the system. By using the process ID as part of a file name that a user has built to be executed, the problem of two execute files of the same name is avoided.

7. mail - command used to send messages. It is used to relay to the RSA (Resource System Administrator), when a cleanup is due for a particular reservation system.

8. sleep - suspends execution for an interval. It is used to execute a command to access a reservation database periodically while the database is in use. Once the command has been executed a specified number of times and access still has not been gained, a message is returned for a later retrial.

9. tgetent - used to extract the entry from the termcap database for the particular terminal being used. This allows the program to utilize the termcap functions and the Resource System Package can then run from any terminal defined in the termcap database.

10. tgetstr - gets the string value of the capability being used and places it into the buffer. The capability used within the program is 'cl' to clear the screen each time a new screen is evoked.

11. tgoto - function from the terminal capability database termcap. It returns a cursor addressing string. This is used to place the cursor at the top of the screen for each new screen displayed.

C. PROPOSAL

ROCHESTER INSTITUTE OF TECHNOLOGY
SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

A RESOURCE SYSTEM PACKAGE THAT ALLOWS A DESIGNER TO DEVELOP RESERVATION SYSTEMS

BY
GARY L. FOLEY

THESIS PROPOSAL

APPROVED BY:

_____

PROFESSOR WARREN CARITHERS

_____

PROFESSOR MICHAEL LUTZ

_____

PROFESSOR PETER LUTZ

JUNE 1, 1983

3. TABLE OF CONTENTS

1. Title Page.
2. Approvals.
3. Table of Contents.
4. Introduction and Background.
    - 4.1 Problem Statement.
    - 4.2 Previous Work.
    - 4.3 Theoretical and Conceptual Development.
    - 4.4 Glossary.
5. Project Description.
    - 5.1 Functional Specification.
        - 5.1.1 Functions Performed.
        - 5.1.2 Limitations and Restrictions.
        - 5.1.3 User Inputs and Outputs.
        - 5.1.4 System Files.
    - 5.2 System Specification.
        - 5.2.1 System Organizational Chart.
        - 5.2.2 System Data Flow Chart.
        - 5.2.3 Equipment Configuration.
        - 5.2.4 Implementation Tools.
    - 5.3 Implementation Plan.
        - 5.3.1 Deliverable Items.
        - 5.3.2 Milestone Schedule.
6. Qualifications.
    - 6.1 Personal Background.
    - 6.2 Courses Taken.
    - 6.3 Books and Articles Read.
    - 6.4 Programs Written.
    - 6.5 Projects.
7. Bibliography.
8. Grading Criteria.

4.1 Problem Statement:

As my thesis project, I intend to implement a generalized resource system that can be utilized to design a reservation system for any type of resource. This would involve a self guided package that a designer would follow to set up his/her desired implementation. Some examples of its use might be:

Terminal Reservations
Room Reservations
Dial-up Line Reservations
Textbook Reservations
Seat Reservations (classroom, special presentations, etc.)

4.2 Previous Work

Relevant to Reservation Systems:

The majority of work done in this area deals with hotel, airline, and theatre reservations. A number of software products to perform these operations are available. This area is relatively new and much research and development is taking place.

In my library search, I used the online databases at the Xerox Technical Library. About thirty sources were cited; one of these dealt with the topic. This article, "Travel Planner", discussed a data base that keeps up-to-date information on a variety of cities such as hotels, car rentals, airlines,and entertainment. Actual reservations are not made through this package. The remainder of the sources cited were not articles , but were, rather, actual products on the market. I also looked through Computer and Control Abstracts where I found nine sources, of which three were in journals available to use. I found six sources in ACM manuals (of which three were available); of these six articles, only very general information about what the particular system or package could do was given. These articles gave me some ideas, but the indepth design of such a package was not covered. Again, these dealt with specific reservation systems such as airline, hotel, or theatre and

not a general reservation system. I was informed that a few universities have developed terminal reservation systems but have not been able to locate any documentation. The research done offered some good ideas for the project, but the design will be implemented from personal background.

4.3 Theoretical and Conceptual Development

The main thrust of this project is the development of an easy-to-use package to setup online reservations of various objects. Through the use of such tools as cursor movement optimization, screen updating, the Mistress database package, and the C programming language, which are all available to the RIT VAX/UNIX * systems, the Resource System Package will be developed.

The Mistress Database will be used to store all the information within a reservation system. A screen formating package will be used to develop the screens for the Resource System so that the designer is able to step through the package. The designer of a particular system will develop a set of screens for the particular reservation system. The entire package will be menu driven to make it easier for both the designer and user to deal with. The C programming language has the ability to tie together all the different parts of the system so that a concise control program can be written to direct the input and output of information.

* UNIX ia a trademark of Bell Telephone Laboratories.

4.4 Glossary

access permission - read and write access definitions, designer may specify what these are to be for the particular reservation system being developed.
attributes - characteristics of the object to be reserved.
automatic backup - a copy of original files to be used if the original file is destroyed.
characteristics - a description of what makes up an object.
cursor movement optimization - package of a C library function which allows the cursor to be moved optimally from one point to another.
designer - the person developing a reservation system.
ending date - the last date that a reservation can be made.
entities - objects to be reserved.
FIFO - first in, first out.
identification - a value that represents a specific user.
menu driven - screens offering choices are displayed for the user.
object(s) - item(s) that is to be reserved.
queuing - entries made placed on a stack (queue) which are completed in a FIFO manner.
reservation system - a specific system developed to reserve some object(s).
reserve - menu option to select if you wish to make a reservation.
Resource System - the package used to develop a reservation system.
screen - the way the terminal screen currently looks.
screen formatting - to design or update a screen.
search capability - the different ways that you are allowed to search the information.
security - how the information is protected.
selection - choice or option.
setup - menu option to develop a reservation system.
slot - a place that an entry can be made.
starting date - the first date that a reservation can be made.
time parameters - when the object is available to be reserved.
tutorial - menu option which describes how to use the system.
user - a person making a reservation.
value - data associated with a characteristic of an object.

5. Project Description

5.1 Functional Specifications:

The following section describes the various functions to be performed by the Resource System.

5.1.1 Functions Performed

The Resource System will be able to:

A. provide online interactive operations - the Resource System will be available on the RIT VAX/UNIX systems which; it will be an interactive program, prompting the user for reservation requests.

B. provide options to the user - a person using the Resource System will be given various options to choose from depending upon the type of request being pursued.

C. support multiusers concurrently - many users will be able to enter the same (or a different) reservation system at the same time.

D. provide easy to use data manipulations - the users and designers will be able to easily create (designer only), access, maintain, and update a reservation system.

E. provide preformatted screens - menu-driven preformatted screens will lead a user or designer through the system. At the designer level, additional screens can be created for a particular reservation system.

F. queue and deliver requests in a FIFO manner - as requests to update or make entries in a reservation system are made, they will be queued if the system is busy and responded to in a FIFO order.

G. provide quick searching capability - a user or designer will be able to easily examine statistics about a particular reservation system by using menu-driven screens.

H. provide feedback for successful and unsuccessful requests - when a request is successfully completed, a message will be returned to verify it. If a request was not completed successfully, a specific message stating why is returned.

I. provide use of cursor manipulation - within any screen, the user or designer will have the availability to move the cursor to any location on the screen.
J. provide security as desired by the designer - each time a reservation system is developed, the designer will be able to choose the level of security for his/her particular system.
K. provide a tutorial for both designer and user - as one of the options, a designer may read the tutorial to find out how to use the resource system or to look up information on a specific topic. A user may read the tutorial for a particular reservation system which is developed by the designer.
L. provide the option of having all old data (as defined by the designer) deleted - this will be available to the designer to help keep clean efficient storage of the information.
M. provide storage of data - all entries in a particular resource system will be stored in relations within that system's database by way of the Mistress database package.
N. provide the designer with the ability to determine what the method will be for identification when making resevation will be - examples: name , logonid.
O. provide update capability - the designer will be able to make changes to the design of his/her reservation system. Users and the designer will be able to make changes to existing reservations within a reservation system.

5.1.2 Limitations and Restrictions

A. To the Resource System (system defined)

A designer is defined only as faculty of RIT.

B. To a particular reservation system (designer defined)

The designer can:

1). limit the access and update capabilities through the security selection mode and user identification.

2). limit the number of reservations made by a user per day or per week.

5.1.3 User Inputs and Outputs

All input and output is done interactively when using the Resource System or a particular reservation system. Two types of input are possible in both the Resource System and a reservation system: first, some screens give a number of options for the designer/user to select from. By entering an S in the space before a selection, the designer or user will cause the screen correlating to that selection to appear. The second type of input possible is to make an entry with specific information. (This will be explained further as the example below is discussed.) At any time within the Resource system, the session may be aborted by entering ↑d. The following are examples of how a session might go for both a designer and a user.

Designer Session - this is an example of how a designer might set up a Room Reservations system. The items in italics are what actually appear to the designer.

To enter the Resource System on VAX/UNIX:

RS

This enters you into the Resource System and will bring up the following screen:

RIT RESOURCE SYSTEM
Graduate Thesis Project by Gary L. Foley

Select one option below by using the arrow keys to move the cursor left, up, down, or right respectively. Enter an S in the space provided before the option of your choice.

__ *Tutorial - how to use the Resource System*
__ *Setup - develop a reservation system*
__ *Update - make changes to your reservation system*

We shall assume that the designer knows how to use the Resource System and will skip the Tutorial at this time. The designer selects Setup:

SETUP

What is the name of your reservation system?

_____*Reservations*

The __ represents where the cursor will be when this screen appears. This input is of character type which includes blanks, underscores, upper or lower case letters, and any digit. It must begin with a letter. The length, as specified in the Tutorial, is up to 31 characters. An error message will be returned if a reservation system by that name already exists. For our example we will develop a room reservations system.

Enter object to be reserved rooms

Enter characteristic(s) of object - hit return twice when done.

characteristic 1 building number <CR>

characteristic 2 room number <CR>

characteristic 3 capacity <CR>

<CR>

A return will put the cursor on the next line. If no input is given and return is entered again, the next menu will appear.

Enter values for each characteristic - hit return twice to go onto the next screen.

building number _____

room number _____

capacity _____

building number _____

room number _____

capacity _____

etc.

Time parameters:
calendar - dates that system is available - use format mm/dd/yy
hours - the hours of a day that reservations can be made
time slots- how the hours of the day are to be divided (ex 30min slots, 1 hour slots, etc.)

Enter calendar dates that the reservation system is available.
_____*starting date*
_____*ending date*

Enter the starting and ending time of the day that the reservation system is available and how the time slots are to be divided.
_____ *starting time (use 24 hour time)*
_____ *ending time*
_____*time slot division*

Enter maximum number of slots that can be reserved consecutively. Zero represents no limit.
____ *maximum number of consecutive slots*

Enter maximum number of slots that can be reserved in one day. (Zero means no limit.)
maximum number of slots per day

Enter identification used to represent a reservation (examples - logonid, name)
_____ *identification*

Select the security access permission for your reservation system.

Read Access	*Write Access*
__ *everyone*	__ *everyone*
__ *designer only*	__ *designer only*
__ *restriction by criteria*	__ *restriction by criteria*

If restriction is selected, a screen appears to further specify what that criteria is.

Screen Formatting

The designer is given a preformated screen to change to his/her liking. As many screens as needed are developed to make searching and updating easier for the user. I will not define the workings of this function at this time to allow flexibility when designing it.

Select an automatic delete choice below.

__ *each day delete the previous day's files.*

__ *each week delete the previous week's files.*

__ *do not delete any information.*

User Session

To enter a particular reservation system on Unix
ressys (the name of the reservation system you want)
example -

ressys room

The following screen would appear:

Room Reservations

Select your option below by using keys f, g, h, or j to move the cursor left, up, down, or right respectively. Enter S in the space before the option of your choice.

__ *Tutorial - how to use the reservation system*
__ *Reserve - make an reservation in the system*
__ *Update - make a change to an existing entry*

For this example we will assume that the user has already read the Tutorial and is familiar with the use of the Room Reservations system.

Make a Room Reservation

Select an option below to find out what rooms are available. Enter S before your selection. If you would like to limit your selection by more than 1 criteria, enter an S in each option you would like to join before hitting return.

__ *rooms available in a particular building*
__ *rooms available with a particular capacity*
__ *rooms available at a certain time and date*
__ *whether a specific room is available*
__ *rooms available a certain date*

Each option selected above would bring up different screens and selecting multiple options would bring up combinations of screens. for example, selection of rooms available in a particular building would bring up a screen to select the building in particular. Then a

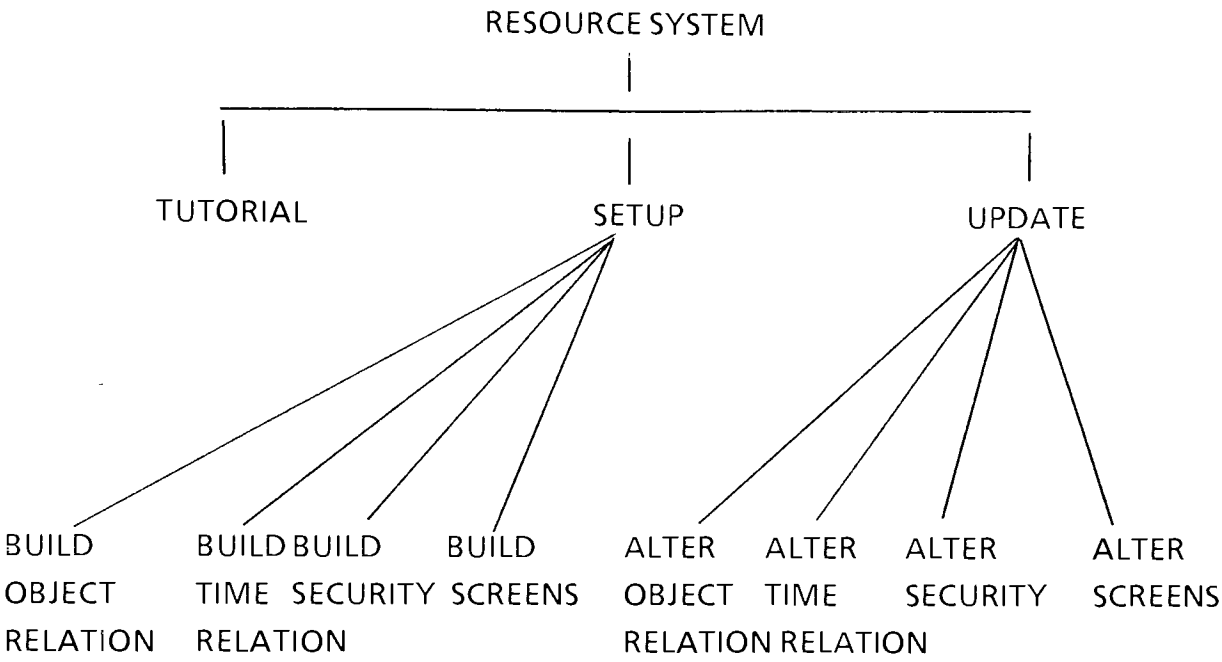
screen following with all the rooms available in that building with the room number and capacity would be brought up. The user would be given options to choose a particular date to see which of those rooms are available and to make a selection or go back to the previous screen or end the session. The other selections act in a similar manner.

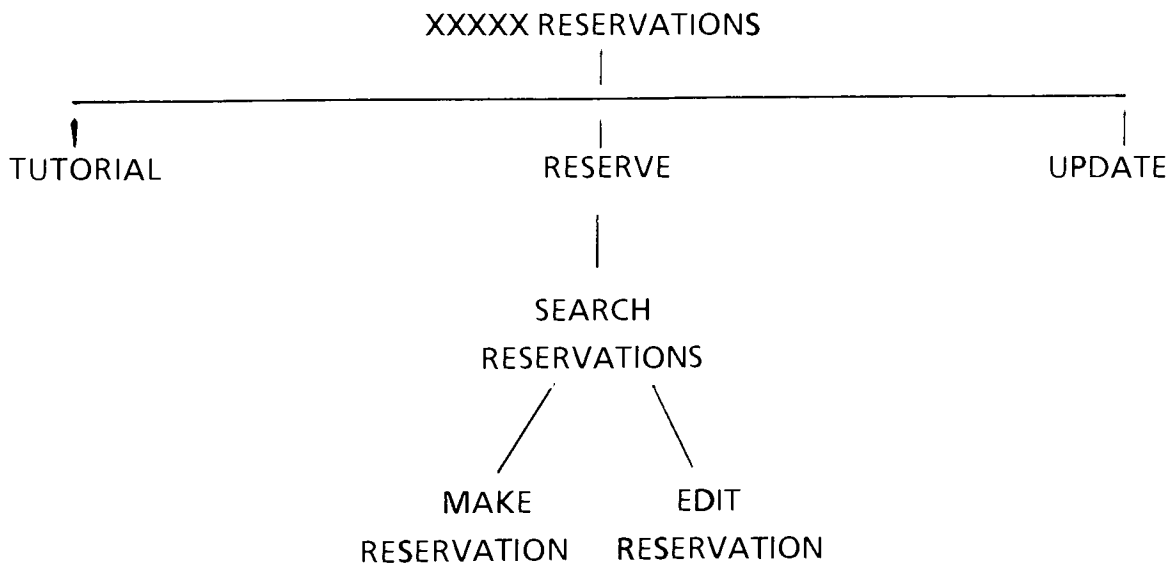
5.1.5 System Files

All information is stored within a Mistress database. A relation or file is created for each object to be reserved containing its characteristics(attributes) and the values of the characteristics. In addition, for each reservation system there will be a time relation or file for each day containing the characteristic(s) that identifies the objects, the times that reservations are possible, and the actual reservations themselves. These files are developed by the designer of a reservation system. These files will be accessed in a variety of ways. A file may be viewed individually or a combination of files may be viewed by the user. The files for a particular reservation system are maintained by the designer of that reservation system.

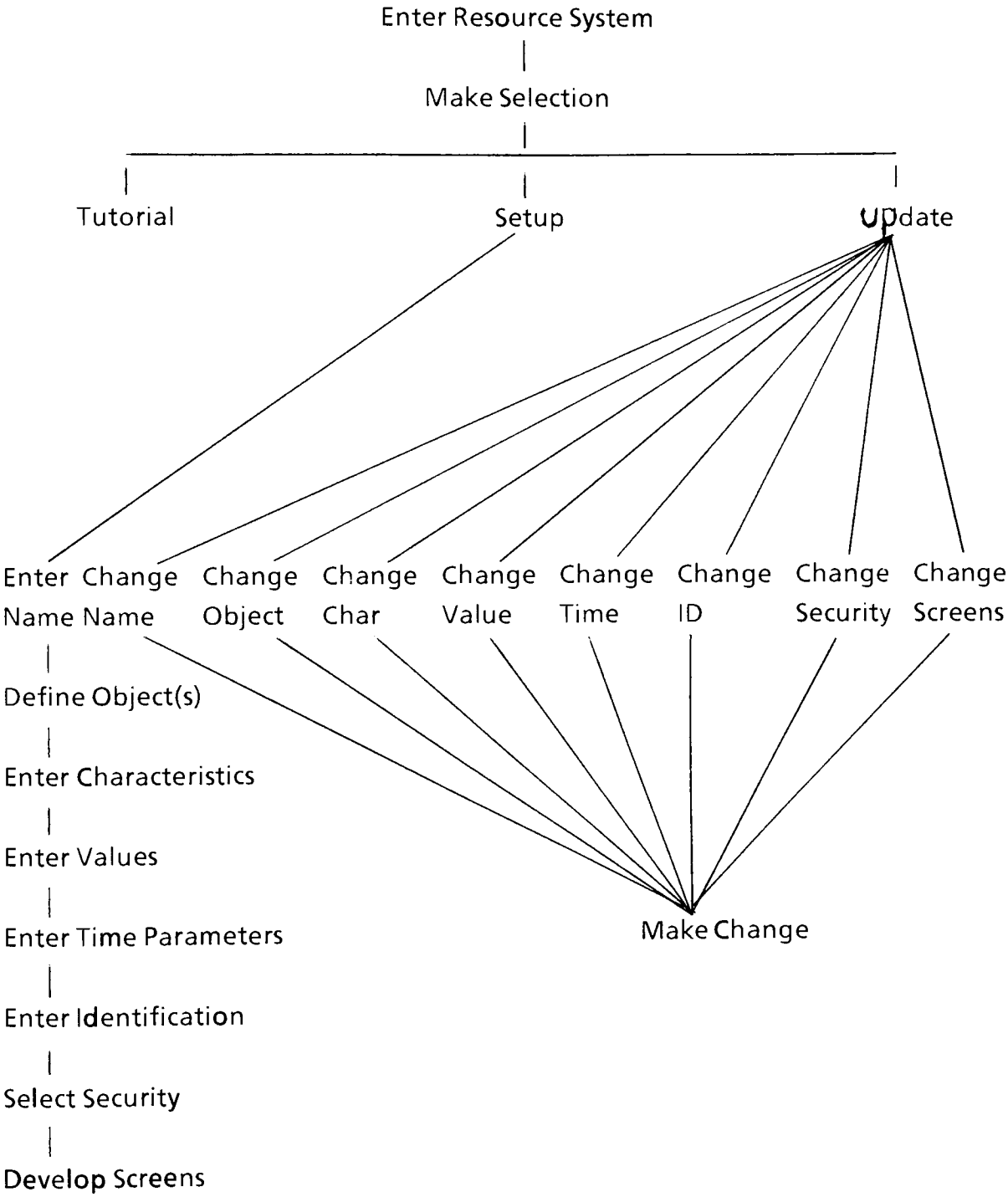
5.2 System Specification

5.2.1 System Organizational Chart





5.2.2 System Data Flow Chart



5.2.3 Equipment Configuration

The hardware to be used for the development and implementation of this project is owned and operated by the Rochester Institute of Technology. This includes RIT's mainframe computers, the VAX 11/780, CRT's and keyboard.

5.2.4 Implementation Tools

There will be a variety of software packages used for the development and implementation of this project. First, the programming language to be used is the C programming language. C is to be used because it offers a variety of data structures which allows the programmer many choices for the best design. Also, C will be used because of its ability to interface with the other software products to be used and the UNIX operating system. The Mistress database package will be used as a storage medium for reservation information. Screen formatting and updating will be used to allow an easy step through process when using the Resource System or a reservation system. In developing the ideas for this project, C's ability to interface well with the above mentioned software will help to make a simple, meaningful, and easy-to-use Resource System.

5.3 Implementation Plan

5.3.1 Deliverable Items

a) program listings - the control programs written in C for the Resource System.
b) supporting documentation for the above programs.
c) designer tutorial - a copy of the online tutorial of how to use the Resource System.
d) user tutorial - a copy of the online tutorial of how to use a reservation system (this only includes general information valid to all reservation systems - the designer will add specifics about his/her particular reservation system).
e) pseudo code - the flow of the control program written in english.

f) example - an actual example of a reservation system built using the Resource System.

5.3.2 Milestone Schedule

June 10, 1983	Thesis Approval
July 8, 1983	Halfway Point Review
August 1, 1983	Thesis Presentation

6. Qualifications

6.1 Personal Background

Work experience - within the computer environment of my job, I utilize clists which are preformated screens in which to make selections or enter required information.

6.2 Courses Taken

Programming - the following are the programming courses taken at RIT. The courses with comments after them are relevant to my thesis project.

210 Program Design and Validation -learned how to manipulate character strings.
305 Assembly Language Programming
320 Data Structure Analysis - learned linked lists and tree structures.
700 Review of Programming
708 Software Architecture
709 Programming Language Theory - learned the C programming language.

Data Base - These three courses were part of my graduate concentration. My thesis project is related to them all since I will be using a Mistress data base for storage.

736 Data Base System Implementation
836 Data Base Systems
846 Information Storage and Retrieval

Other

420 Data Communications Systems
610 EDP Auditing
706 Foundations of Computing Theory
720 Computer Architecture
740 Computer Communications Networks

6.3 Books and Articles Read

See Bibliography

6.4 Programs Written

Mistress - Patient-Surgery Data Base - familiarized myself with the capabilities and workings of the Mistress Data Base.

C - program written to read a C program and sort the code words.

Pascal - various programs written using linked lists, trees, stacks and queues, and variable length character strings.

6.5 Projects

Glorious University - designed a data base for a university containing student and college information.

Swim Club DataBase - designed data base for swim club information.

7. Bibliography

Arnold, Kenneth C.R.C.. "Screen Updating and Cursor Movement Optimization: A Library Package", University of California, Berkeley.

Busch, David D.. "Travel Planner". Interface Age. May 1981. pp. 102-104.

"Computerized Registration Eases Convention Line Lag, Aids Information Queries". Computerworld September 1, 1980. pp. 14-21.

Graham, Neil. Introduction to Programming. West Publishing Company. 1980.

Kernighan, Brian W. and Dennis M. Ritchie. The C Programming Language. Prentice-Hall, New Jersey. 1978.

"Mistress: The Query Language". Draft Version 2. Rhodnius Incorporated. 1981.

"Reservation Facility Links 2,500 Hotel Chain". Computerworld. May 5, 1980. pp. 14,50.

Roberts, Wayne. "Travel Business Gets Boost From Computers". Compdata. February 1980. pp. 5, 34-35.

Tenenbaum, Aaron M., and Moshe J. Augenstein. Data Structures Using Pascal. Prentice-Hall Incorporated., New Jersey. 1981.

Tsichritzis, Dionysios C. and Frederick H. Lochovsky. Data Models. Prentice-Hall, New Jersey. 1982.

8. Grading Criteria

The resource System project is to be graded on the following with corresponding weights:

Criteria	Percentage
Operating Efficiency	20
System Design	50
Correctness of Implementation	20
Supporting Documentation	10

Permission to Reproduce

Title of Thesis: A Resource System Package for Reservation Systems

I Gary Foley, hereby grant permission to the Wallace Memorial Library, of RIT, to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Gary L. Foley 10/31/83