DESIGN CONSIDERATIONS IN THE DEVELOPMENT OF USER-FRIENDLY INTERFACES

by

LESLIE A. EATON

B.S., University of Illinois, 1978

A MASTER'S REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1986

Approved by

[Signature]
Major Professor
# Table of Contents

List of Figures ................................................................. iii

Chapter One: Introduction .................................................. 1
1.1 Introduction ............................................................... 1
1.2 Human Factors and the Perception of User-Friendliness .......... 2
1.2.1 Expectation ............................................................. 3
1.2.2 Control ................................................................. 4
1.2.3 Consistency ............................................................ 5
1.2.4 Short-term Memory .................................................. 6
1.2.5 Closure ................................................................. 6
1.3 Interface techniques .................................................... 6
1.3.1 Menus ................................................................. 7
1.3.2 Forms ................................................................. 11
1.3.3 Prompting ............................................................. 12
1.3.4 Command Languages ................................................ 13
1.3.5 Icons ................................................................. 14
1.3.6 Windows .............................................................. 16
1.3.7 HELP Facilities ..................................................... 18
1.4 Conclusions from the Literature ........................................ 20

Chapter Two: Requirements for the Real Estate Investing Application ........................................ 23
2.1 System Requirements .................................................... 23
2.2 Real Estate Investment Functionalities ............................... 26
2.2.1 The Income Property Statement ................................. 27
2.2.2 Other Application Functions ...................................... 30

Chapter Three: The Design for the Real Estate Investing Application ........................................ 34
3.1 System Architecture ..................................................... 34
3.2 The Design of the User Interface ...................................... 37

Chapter Four: Implementation of the Real Estate Investing Application ........................................ 49
4.1 Using dBASE III Plus .................................................... 49
4.2 Implementation of the Real Estate Investing System ............... 50
4.3 Using the Real Estate Investing System .............................. 51

Chapter Five: Summary ....................................................... 53
5.1 Conclusions .............................................................. 53
5.2 Future Extensions ....................................................... 53

Bibliography ................................................................. 55
Appendix A - Examples ...................................................... 60
Appendix B - Menus and Screens .......................................... 64
Appendix C - Source Code ................................................ 91
List of Figures

| Figure 2-1. | Screen Format | page 33 |
| Figure 3-1. | System Architecture | 39 |
| Figure 3-2. | Property Description | 40 |
| Figure 3-3. | Purchase Terms | 41 |
| Figure 3-4. | Income | 42 |
| Figure 3-5. | Expenses | 43 |
| Figure 3-6. | Ownership | 44 |
| Figure 3-7. | Tenants | 45 |
| Figure 3-8. | Rental Terms | 46 |
| Figure 3-9. | E-R Relationship Diagram | 47 |
| Figure 3-10. | Main Menu | 48 |
| Figure A-1. | Initial Screen | 64 |
| Figure A-2. | Main Menu | 65 |
| Figure A-3. | Property Identifier Selection | 66 |
| Figure A-4. | Income Statement - Page 1 | 67 |
| Figure A-5. | Income Statement - Page 2 | 68 |
| Figure A-6. | Income Statement - Page 3 | 69 |
| Figure A-7. | Income Statement - Page 4 | 70 |
| Figure A-8. | Property Description Selection | 71 |
| Figure A-9. | Property Description | 72 |
| Figure A-10. | Purchase Terms Selection | 73 |
| Figure A-11. | Purchase Terms | 74 |
| Figure A-12. | Ownership Records Selection | 75 |
| Figure A-13. | Ownership | 76 |
| Figure A-14. | Tenant Record Selection | 77 |
| Figure A-15. | Tenant Records | 78 |
| Figure A-16. | Lease Terms Selection | 79 |
| Figure A-17. | Lease Terms | 80 |
| Figure A-18. | Income Records Selection | 81 |
| Figure A-19. | Income Records | 82 |
| Figure A-20. | Income Reports Selection | 83 |
| Figure A-21. | Income Reports Year Selection | 84 |
| Figure A-22. | Income Report | 85 |
| Figure A-23. | Expense Records Selection | 86 |
| Figure A-24. | Expense Records | 87 |
| Figure A-25. | Expense Reports Selection | 88 |
| Figure A-26. | Expense Reports Year Selection | 89 |
| Figure A-27. | Expense Report | 90 |
Chapter One: Introduction

1.1 Introduction.

The quality of the human-computer interface is a main determinant in the success and the degree of use of a computer-based interactive system [Ben84]. In addition to the computer professional, a new class of users has emerged due to the proliferation of interactive systems in the office, home, and school. These new users differ in ages, backgrounds, interests, and abilities. Unlike computer professionals who are willing to cope with the idiosyncrasies of terse command languages and computer jargon, these new users often don't have the patience or the inclination to do so [Ben84].

This new class of users is the motivation behind the development of user-friendly interfaces. One problem in developing user-friendly interfaces lies in the diversity of user personalities and the diversity of situations in which the systems will be used. This diversity makes it difficult to prescribe a universal user-friendly interface paradigm.

This work will look at some human factors which affect the user's perception of user friendliness. It looks at
some common interface techniques and some items that interface designers should consider when designing a user-friendly interface. It also demonstrates these techniques in the requirements and design for a real estate investment database application for a personal computer.

1.2 Human Factors and the Perception of User-Friendliness.

Proposed sets of guidelines to use when developing user interfaces tend to reflect an author's bias towards specific human factors. These sets of guidelines offer conflicting advice on how to design a user interface and make it "friendly". The goal in designing a user-friendly interface is to seek a workable compromise between conflicting design guidelines in order to create an interface that the user finds easy to use, easy to learn, is not intimidating to the user, is enjoyable to use, adapts as the user becomes more proficient, and provides help when the user is in trouble [Lars82].

A few of the human factors problems the interface designer should consider are expectation, control, consistency, short-term memory, and closure.
1.2.1 Expectation

The user's attitudes and experiences will provide him/her with certain expectations. Users bring their expectations of human to human dialog to the computer session. For example, the user has ideas on how long he/she should wait before receiving an answer to a question, what to do if the answer does not fit the question, and how to tell when the conversation is over. These human conventions for conversation provide a standard by which the computer session is judged [Reit84].

Feedback is important to the user. A user may panic if the screen goes blank or unexpected things appear on it [Pete79]. Users may also panic if there are unexpectedly long delays. In one study on response time, users preferred a consistent response time to each inquiry rather than a varying response time - even if the consistent response time produced a greater average wait time than the average of the variable response time [Shne79]. Myers suggests using "percent done" or another type of dynamic progress indicator to reassure the user that the system is still processing the user's task [Myer85].
An unnatural interface is one that does not support the user's initial or habitual response tendencies and this may lead to confusion for the user [Whit85]. Information requested or displayed by the system should match the user's mental representation of the information. Data should be presented to the user in the way the user is accustomed to handle it. The vocabulary should be the user's vocabulary; the syntax should follow the syntax the user is accustomed to [Lust85, Reit84].

1.2.2 Control

In order to feel comfortable in interacting with a computer, the user needs to feel in control. Any activity of the system should be recognized by the user as a direct consequence of his/her actions [Gain81]. Novice users may prefer a computer-guided system with extensive help and error facilities; experienced users may prefer a user-guided system which accepts short messages and allows shortcuts to speed up the dialogue [Thim80]. However, as users gain experience, there is evidence that they perceive computer dominance with resentment [Shne79]. The Library of Congress recognized this and changed the prompt from the computer commanding "Enter next command" to the more neutral "Ready for next
command" [Shne79]. The challenge for the interface designer is to develop an interface that evolves and gives more control to the novice as he/she gains more proficiency.

1.2.3 Consistency

Inconsistency in a user interface will cause confusion and frustration for the user. If an action which works in one part of the interface causes an unexpected reaction when used in another part, the user's performance will decrease as time is spent recovering from these unexpected results. The user's memory is also stressed as he/she tries to remember which actions to take in different parts of the interface [Gain81, Lust85]. Lustman's guidelines [Lust85] for interactive information systems require that the user interface be contained in one single design module to reduce the inconsistencies users are sometimes presented with and that, regardless of which part of the system the user interacts with, the same command should be required to achieve the same result. Gaines [Gain81] states that the same terminology and operational procedures should be uniformly available and consistently applied throughout all system activities.
1.2.4 Short-term Memory

Experiments have shown that there is a psychological limit of 7 plus or minus 2 items that a person can retain in their short-term memory. Memory aids should be provided to the user to help him/her keep track of variables that change often; to remind the user where he/she is positioned in long, complicated, or nested procedures; and to remind the user of which actions can be taken next [Reit84].

1.2.5 Closure

Closure is defined as the relief that a person has when he/she can release the items held in short-term memory at the completion of a task. The need for closure creates a powerful desire in the user to complete a task, reduce memory load, and gain relief [Shn79]. Shneiderman suggests that the need for closure should be reflected in how a task is structured. Interface designers should allow the user to complete a large task by completing small component tasks.

1.3 Interface techniques

Interface techniques that are currently in use include
menus, forms, prompting, command languages, icons, windows, and HELP facilities.

1.3.1 Menus

Both Shneiderman [Shne80] and Martin [Mart73] have stated that novice users normally require a menu-driven interface. The appeal of a menu-based system is that the user does not have to recall the command in order to use it, only to be able to recognize it from a displayed list of choices. Menus are widely used but there is still little systematic knowledge on how they should be structured.

There are a number of different variables the designer should consider when designing a menu-driven interface. These include how to structure the menu system, how to navigate within the menu system, how many choices to put on a menu, how to order the choices, deciding what each menu category should contain, how to name the categories, and when to have the menu appear on the display.

One of the first decisions is whether the menu system will be single-leveled or hierarchical. Allen has shown that depth (the number of levels into a hierarchically
organized menu required to reach the "target" location) and level (the number of menus between the root menu and the menu on which the response was made) have an effect on the time the user takes to make a response [Alle83]. Navigation problems may occur in hierarchical menus if the user loses his sense of position within the system [Sava81]. One suggestion is to provide a memory "trail", reminding the user what the current level is and the choices he/she made to get there [Reit84].

The designer should also consider whether the system will allow an experienced user to take shortcuts. One option is allow the user to enter all menu choices at one time for the intermediate menus between the current level and the target level. Another option is the ability to back up to previous levels and continue processing instead of returning to the root menu after each target has been reached.

The decision of how many choices to place on a menu should take into consideration the number of keystrokes the user will need to make his/her selection, how long it takes to scan the menu, and how dense the display will appear. Petersen feels that the menu should contain no more than nine items; they should be numbered 1 - 9 so that the user only has to enter one keystroke.
when making his/her selection [Pete79]. While the short-term memory limit may not be a factor with menus, there is empirical evidence that people prefer shorter menus [Alle83, Sava81].

The designer should also consider how to organize the choices. Petersen [Pete79] states that the menu items should be functionally organized, with alphabetical organization being the second choice. In a study in which sets of commands were functionally, alphabetically, and randomly organized, Card [Card83] concluded that an alphabetic arrangement was the fastest for the user to use until there had been repeated practice with the menu. At that point, all organizations were equivalent in terms of user performance. There is some evidence that the most frequently used choices should appear first in the menu since users typically do not look at all choices before making their selection [Alle83].

The user should only be presented with choices that are valid, the list of choices should exhaust all possible actions, and the choices should be distinct (i.e., non-overlapping) so the user experiences no confusion when making his/her selection. Success in using menu-based information retrieval systems depends on
knowing what is included in the category or partition of menu choices [Duma83]. Having an expert pick out the
name of a menu category is one such method; giving examples of what is in the category is often more
helpful to the user. For example, instead of giving the menu category the name "Operating Expenses", a list
such as "Utilities, Cleaning, Maintenance" would be given.

Another decision the designer must make is when to have a menu appear. Two common strategies are to use
pull-down or pop-up menus. The pull-down menu remains on the screen as the user changes levels in the hierarchy,
usually changing contents to reflect the choices available at each new level. In most cases, a certain
area of the screen is designated as the menu area and is only used for displaying the menu. The pop-up menu
generally is not displayed unless the user requests to see it. In some implementations, pop-up menus either
cover-up what is already displayed on the screen or cause the current display to be rearranged. Little study
has been done on which of these two methods is preferable, especially in relation to the ideas of
positional constancy [Teit83]. Positional constancy allows the user to develop a "map" of the display which
promotes expectations for the location of future
information.

Menus in themselves don't guarantee a user-friendly system. Badly chosen design options will adversely affect the user's perception and pleasure in using the system [Whit85].

1.3.2 Forms

Using forms as an interface technique tries to take advantage of the user's experience in dealing with paper forms and making the analogy that filling out a computer form is the same familiar process [Luo81]. Some advantages in using a forms interface lie in it's pictorial nature, which can use the placement and grouping of data to convey meaning to the user [Lars81]. The fields on the form are seen and understood in the context of the entire form. This can prevent confusion for the user [Ingr85].

Forms are another version of a question and answer dialogue between the user and the computer. In contrast to other dialogue techniques, the user can change items on a number of fields on the form before completing the form. The designer still must exercise judgment in designing a forms interface. Data should be requested in
the same format as the user is accustomed to seeing it,
e.g., for Americans, the system should ask for
month-day-year instead of day-month-year [Whi85]. The
user should not be expected to enter data which the
system can provide (or which the user has provided once
before). Forms can make life easier for repetitive
dialogues by providing default values for fields and
using history from previous forms to partially fill out
the next form for the user [Ingr85]. In a form system,
the designer must also decide when feedback should be
given to the user - should he/she be corrected as each
field is entered, or as the complete form is entered,
or when there is enough information on the form for the
system to recognize that there is a problem?

1.3.3 Prompting

Prompting is one of the most familiar question and
answer dialogue techniques between the user and the
computer. The prompt is the text that asks the question
and at the same time, specifies the type of information
that the user needs to supply to the system. One of the
advantages to this interface method is that it is one of
the easiest techniques for a user to learn. The prompts
can be made as detailed as necessary to lead the user
through the dialogue [Pete79]. However, the advantages
this technique has for the novice user become disadvantages as the user gains experience. The dialogue is sequential. A detailed dialogue will be slow to complete. Generally, there are no provisions for speeding up the dialogue or for changing a previous choice before completing the entire dialogue. This rigidity in the structure of the dialogue can lead to frustration for the novice and boredom for the experienced user.

1.3.4 Command Languages

Menus, forms, and prompting guide the novice user through the system. In each of these interface techniques, as the user gains more experience with the system, the user will look for shortcuts to speed up the dialogue. In menu-driven (or form-driven or prompt) systems which build command lines, one shortcut is for the user to directly enter the system command he/she wishes to execute.

How does the user learn the command language? Larson suggests a teaching method for menu-driven systems that allows for varying user proficiency. This technique involves segmenting a portion of the screen for use as the "command" window. As the novice selects from menus,
his choice is copied to the command window. Successive menu choices are applied to the contents of the window until the complete command is formulated. With experience, the user can enter his choices directly into the command window or use the menus to help him choose. This approach allows the experienced user to bypass the menus and teaches him how to formulate commands [Lars82].

1.3.5 Icons

Icons are one of the newer interface techniques to appear. An icon is a pictorial representation used to identify an object or a command involving an object. Hemenway [Heme81] has identified three types of command icons. One type is icons which depict objects that the corresponding commands operate on ("command objects"). An example of this type of icon is the canister, commonly seen on flowcharting templates, used to represent a file. The second type of icon is one that represents the operation that the command performs ("command operations"). An example of this type is the scissors, used in the Draw software package, to indicate that the user can "cut" or dash/undash lines. The third type of icon represents both objects and operations. An example of this icon type is the "X" superimposed over a
box, used in the Superpaint software package, to represent deleting an image area.

Icons can be effective because they are visually more distinctive from each other than is the printed word. They can represent a lot of information in little space. They can clearly represent variations of some dimensions (e.g., length, width, brightness) by visually displaying the variation.

There is a two-step process the user goes through when first using an icon. First the user must determine what the icon represents. Then the user must perform a mental "link" between the icon and the corresponding command. "It's easy to determine what an icon depicts if the user is familiar with the object or symbol depicted, and if the depiction is clear and unambiguous, and if it's a conventional symbol or caricature of a common, familiar object" - Hemenway thus provides a statement full of caveats for the interface designer [Heme81]. If there is no existing icon that matches this description, a new icon may need to be defined through analogy. The quality of the analogy will determine if the link between icon and meaning is easy to discover and whether it is retained in the user's memory.
Icons are most commonly used in menu systems together with a mouse or other pointing device. Whiteside measured the performance and subjective reactions of users completing a standardized file manipulation task using seven different software systems which used command, menu, and iconic user interfaces. In this study on different user interfaces, users experienced difficulty in using both of the iconic interfaces. Users were confused not only with the meaning of the icons but also with how to use the mouse input device [Whit85]. The designer of an iconic interface should consider how the user will learn to point to the icon and not to a component of the icon or to the area surrounding the icon [Gala83]. Mouse technology is currently implemented inconsistently between software packages. Each software package uses a different button control sequence with the mouse. Systems read cursor positions of the mouse differently. These inconsistencies prevent the user from drawing on previous experiences with mouse systems when he/she uses a new system [Warf83].

1.3.6 Windows

Another interface technique is to provide the user with windows, which are separate areas or partitions on the display screen. Windows have been proposed to match the
user's experience of a desk top. They are designed to help the user integrate the results of subtasks in order to complete a larger task. The user's short-term memory burden is theoretically reduced because all the materials the user must reference are right on his "desk top" [Kons85]. In their simplest form as a partition of the display screen, windows have also been used to provide clocks, status indicators, fixed menu areas, and help and status windows [Grei84].

When designing a window system, the designer should consider that windows can quickly become too small to adequately display information [Warf83]. Most windowing systems require high resolution graphics cards to produce reduced images which are an added expense to the user. If the windowing system is designed as an integrated package, the designer should be concerned with consistency between the components of the system [Warf83]. The concern with a windowing system that uses existing packages is whether or not it also provides a consistent interface to each of these packages. Users will get confused and frustrated switching between packages that have different interfaces and commands to perform similar functions within the packages. In an integrated system, the designer must be concerned that the user does not perceive inconsistencies in usage.
because he/she is using different types of packages, such as a word-processor, spreadsheet, and graphics package. Unless the user has control over window placement, the designer should also consider the effects of positional constancy as windows appear and disappear on the display.

1.3.7 HELP Facilities

Most computer systems offer some form of HELP facility. An appropriate design of a HELP facility is important to the user's success in interacting with the computer. The designer should be concerned with who initiates the request for help, what type of help should be given, and how much help should be displayed.

In Cohill's study with novice users, he concluded that it was more effective if the user initiated the request for help rather than the system initiating the help[Co85]. In addition to giving the user a greater sense of control, Cohill found that user performance remained at a more basic skill level when the computer had partial or complete control of initiating the help request.

The designer may want to consider different kinds of
help to give the user. Most HELP systems respond to the
user with an explanation of the types of responses the
system will accept as valid. Fischer [Fisc85] looks at
HELP systems as either passive or active. Passive HELP
facilities provide details about syntax, terminology, or
system functions without placing the help information in
the context of the task the user is performing. Active
HELP facilities are meant to explain and guide the user
through the task by recognizing what the user is trying
to do, evaluating the user's performance, and deciding
when to interrupt and how to help the user when the user
is performing at a suboptimal level.

Reed [Reed81] classified problem solving strategies as
either error-preventing or error-correcting. He found
that when users start to solve a problem, they already
have an idea on how to arrive at the solution based on
their past experiences. These "first" solutions are
revised and debugged until they are an "error-corrected"
solution to the problem. Reed concludes that
error-correcting is a natural process for the user;
conversely, the error-preventing HELP facilities of most
computer systems are unnatural to the user. Reed's
suggestion is to provide the user with templates of the
system operations. These templates will contain default
values, automatic constraint-checking on substituted
values, and an interactive segment debugging facility to allow the user to debug subsets of the complete operation.

The designer must also consider how much help to give the user. Whiteside's experiments with different interface styles showed that user performance increased when new users were insulated from commands they wouldn't immediately need [Whit85]. When they were later guided to these commands, users found actual examples of command-line syntax useful. Whiteside also found that when users were not referred to specific HELP items, they were often overwhelmed by the amount of help available and spent time reading HELP which was not pertinent to the task at hand. On a more practical level, Cohill found that users preferred not to have the HELP messages erase the screen area [Cohi85]. This way, users could compare their work side by side to the text of the HELP message.

1.4 Conclusions from the Literature

The field of research to define user-friendly interfaces is in its early stages of development. The literature varies as to what makes a user-friendly interface. Many authors describe their personal preferences without
sufficient research to back their opinions; others base conclusions on research of such small scale as to make the results questionable. The research that has been done raises as many questions as it answers. There is still the need for more research in this area.

Friendly user interfaces are being demanded by new computer users. The proliferation of personal computers and computer usage by non-computer professionals has also created a great laboratory in which the market place will help define the interface techniques that satisfy users. Many of the techniques and ideas described in this work are the result of relatively new research, however they are already being incorporated into the newest computer products.

Success in designing a user-friendly interface depends on having a complete picture of the user - his/her purpose in using the system, experience level and ability, learning style, and personal preferences.

The following chapters will describe the design and implementation of a real estate investment application for a personal computer. The user interface to the
application will demonstrate several of the user-friendly techniques described previously.

Chapter 2 will describe the overall requirements for the application program. These include the requirements the system will meet to achieve user-friendliness as well as the system functionality requirements. Chapter 3 will present the design of the application interface. Chapter 4 will present the implementation of the interface and the underlying application. Chapter 5 will provide a summary of the project, conclusions, and future extensions.
Chapter Two: Requirements

This chapter contains the requirements for a real estate investment application on a personal computer.

2.1 System Requirements

The operating environment for this application will be an IBM-compatible personal computer running MS-DOS 3.0 and the dBASEIII PLUS database management system package.

The real estate investment application will use the dBASEIII PLUS database, but the user does not have to be familiar with the dBASEIII PLUS package to use the real estate investment application. The application will provide the user with a menu interface but will also allow the user to use dBASEIII PLUS commands directly on the application database.

Following guidelines in Lustman [Lust85] and Teitelbaum [Teit83], the user display will always have the same structure. (See Figure 2.1.)

Area 1 displays menu and function identification information such as menu name, menu level, and system
function.

Area 2 is the working space. It displays either a menu or the input/output data from an application function.

Area 3 displays the dBASEIII PLUS command that is currently operating.

Area 4 displays control information such as instructions for moving between menu levels, how to access HELP information, and system error messages.

The menu interface is organized as a tree structure with the main menu located at the root of a tree. The menu organization will allow for forward and backward navigation through the tree. The user can return to previous menus by backing up one menu at a time or by naming the menu he/she wants to return to. The user can enter choices to move forward more than one menu at a time. A single command or a function key will allow the user to return to the main menu from any level in the tree.

The menus will direct the user to either input forms or a menus to produce reports. If the target destination is an input form, the user will be entering data by
filling out forms. Explanations of the fields on these forms will be provided through HELP functions. The input from these fields will be collected in a buffer and will be entered into the database only when the data for a complete record has been collected. If the target destination is an output form, the user will have the choice of viewing the report on-line or printing it in hard-copy form.

The user will be allowed to add, update, and delete records. If the user is deleting a record, the user will have the option of recalling that record before he/she ends the computer session. When the user begins the process of ending the session, if a record has been deleted, the user will be warned and will be given one more opportunity to recall a deleted record; otherwise, the deleted records will be purged from the database at the end of the computer session.

Response times will vary with the complexity of the processing function. A busy indicator, such as the word 'BUSY' flashing on/off, will signal internal activity until there is detectable activity on the screen.

The user may select to work at different proficiency
levels. At the beginning level, the user will interact with the system through the menu interface. The dBASEIII PLUS commands that the system is using will be displayed on the screen for the user to see. HELP will be system activated. At the intermediate level, the user will interact with the system through the menu interface or by entering commands directly. The user can turn off the system activated HELP but can still issue requests to HELP. At the advanced level, the user can choose to interact with the system using dBASEIII PLUS commands or their approved abbreviations.

HELP facilities will be available to the user. There will be HELP to give an overview of each menu, to explain real estate terminology, to give command syntax and a simple example of command usage, to explain each field and its domain on the input forms, and to explain navigation through the system. When HELP is displayed, it will not erase the target of the HELP.

2.2 Real Estate Investment Functionalities

The user will be able to use this application to produce an Income Property Statement which can be used to determine if a property is a good investment. The user will also be able to store property description,
purchase terms, co-ownership, income, expenses, and tenant information for properties that he/she owns. Real Estate terminology and calculations will be described in the implementation as they are used.

2.2.1 The Income Property Statement

An Income Property Statement allows the user to analyze the income potential and expenses for a property. To prepare the Income Property Statement, the user will be asked to supply

- property description information: property address, number of units
- asking price
- purchase price (The user has the option of leaving this information blank. The program will calculate a purchase price which will give the user a cash flow = 0, which will be considered the minimum amount for positive cash flow.)
- down payment (The user may enter a specific dollar amount or may specify down payment as a percentage of the purchase price.)
- an income grid: number of units and the actual rent amount/month, the fair-value rent amount/month

- any other income the property generates

- vacancy/credit loss factor (The user has the option of entering this as a dollar amount or as a percentage.)

- fixed expenses: real estate taxes, insurance

- operating expenses: these are 12 of the 14 categories used by the Internal Revenue Service on Form 1040, Schedule E - Supplemental Income Schedule. Real estate taxes and insurance are the other two categories.

- loan information: amount of 1st and 2nd mortgages, interest rates, and length of loan. The application will only handle fixed rate mortgages at this time.

The system will calculate
total monthly actual rent
total monthly fair-value rent
annual gross income
scheduled gross income
effective gross income
total annual expenses
annual net income
principal and interest on the 1st and 2nd mortgages
cash flow
annual depreciation (At this time, the application will only provide straight-line depreciation.)
equity build up for the first year of ownership.

The user will be able to model investment situations using the Income Property Statement. The system will allow the user to enter "speculative" data, i.e., data for the purpose of analyzing different investment situations. The user has the option of storing this data for future use. The system will store this data as speculative data until the user deletes it or until the user marks it as actual data. An example of the user wanting to change speculative data into actual data is when the user purchases one of the properties for which he/she has created an Income Property Statement. This
will save the user the effort of re-entering that data. Only actual data will be used to prepare the output reports 2 - 9 that are described below.

The user can also prepare an Income Property Statement for any property that he/she already owns. In this case, the user specifies which property he/she is interested in viewing and the application will create the Income Property Statement from the information in the database.

An example of an Income Property Statement is included in Appendix A. Data that the user supplies is highlighted. This example will be available online for the user to examine. Explanations of the Real Estate terminology used will also be available online.

2.2.2 Other Application Functions

For property the user already owns, he/she will be able to add, update, and delete

- property description information: property address (street, city, state, zip), number of units
- Ownership information: the property the user owns

- Co-owner information: co-owner name, co-owner address (street, city, state, zip), percentage of ownership, which properties he/she co-owns with the user

- Property purchase terms: purchase price, down payment, closing date, possession date

- Property mortgage information: mortgage amount, mortgage terms, length of mortgage

- Property income information: income type (rent, interest from security deposits, the security deposits themselves, other sources), date received, amount received, payee, comments about this transaction.

- Property expense information: expense type (classified into the 14 IRS categories, date paid, amount paid, payee, method of payment (check, cash, credit card), comments about this transaction.
Examples of these types of information are also included in Appendix A.

Output reports available to the user are

1. Income Property Statement
2. Summary of income by property, by month
3. Summary of income by property, by year
4. Summary of expenses by property, by month
5. Summary of expenses by property, by year
6. Summary of expenses by property, by year, by type
7. Summary of cash flow for a selected property, by month
8. Complete listing of income for a selected property, by year
9. Complete listing of expenses for a selected property, by year
<table>
<thead>
<tr>
<th>Area 1</th>
<th>Menu and Function Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area 2</td>
<td>Working Space</td>
</tr>
<tr>
<td>Area 3</td>
<td>Currently Executing dBASEIII PLUS Command</td>
</tr>
<tr>
<td>Area 4</td>
<td>Control Information</td>
</tr>
</tbody>
</table>

Figure 2-1. Screen Format
Chapter Three: Design for the Real Estate Investing Application

This chapter will describe the system architecture of the Real Estate Investing System. It also will present the design for the interface to the Real Estate Investing System.

3.1 System Architecture

The Real Estate Investing System is designed as a modular system. An analysis of the investing process showed that there would be four categories of system usage [Beck82]. These categories are property, rental, income, and expense information. These categories not only contain different types of information but the user will generally perform operations and access the data in these categories at different times in the investing process.

The property information category includes preparing the Income Property Statement. Income Property Statements are generally prepared on an as-needed basis, e.g., the user is looking into purchasing a property and would like to evaluate its profitability, or the user would like to determine if an investment he/she already owns
is profitable.

The property information category also includes adding, updating, or deleting the property description, purchase terms, and ownership data. Adding data corresponds to the user's decision to purchase an investment property, deleting data corresponds to the user's decision to sell an investment property. In terms of database operations, selling a property will be considered the deletion of both the purchase term and ownership information for a property from the database. Property description information will remain until the user removes the associated income and expense information from the database. Few updates are done on data in this category. Examples of when updates would be made are the user decides to turn his/her residence into an income property, sell some of the assets of a property, or refinance a property.

The rental information category includes tenant and lease term data. The data in this category will be added as new tenants and leases are made for newly purchased as well as existing property. Deletions will occur as tenants move out or when properties are sold. In terms of database operations, selling a property (i.e., deleting the purchase term and ownership information)
will cause a triggered delete of lease term and tenant information in the database. The data in this category will change as tenants move in and out, leases are renewed, and rents change.

The income and expense information categories will be the most volatile areas in the database. New income records will be added to record monthly income; additional expense records will be needed as bills are paid. Updates to income and expense records already in the database will be infrequent. Deletions of income and expense records are allowed. In terms of database activity, in order to delete a property description from the database, the user will first have to perform deletions of all income and expense records for that property. The rationale behind having the user explicitly delete these records (instead of performing an automatic triggered delete) is a defensive move so that the investor will not be able to unwittingly delete records that may be needed for income tax purposes.

These four categories - property, rental, income, and expense - are represented in seven permanent tables in the database. There is also a transient storage area in the system to store the Income Property Statement data.
Figure 3-1 is a diagram of the system architecture.
Figures 3-2 through 3-8 display the data field formats for the seven permanent tables in the database. Figure 3-9 is a diagram showing the relationships between the data items used in the system.

3.2 The Design of the User Interface

The interface will primarily be a menu-driven interface in order to accomodate the novice user as well as the proficient user [Shne80]. The user will first interact with the application through an initial main menu screen. The user has the option at this point to signal to the system that he/she will be interacting with the system by using dBASEIII plus commands. The main menu (Figure 3-10.) allows the user to select one of ten choices. Each of these ten menu choices will lead the user to another menu level. Although there are ten choices, the choices are labeled so that it will still take only one keystroke to enter the selection.

For most choices, the next menu level will allow the user to add/update/delete/list records in the database. The user will be provided with a form to help him/her format a request. The exceptions are choices a., h., and j. If the user selects choice a., the next level
displays the Income Property Statement form. If the user selects choices h. or j., the next menu displays the report types that are available.

The complete set of interface menus are provided in Appendix B.
Figure 3-1. System Architecture
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Characteristics</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>property_id</td>
<td>character, length = 6</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>property_street</td>
<td>character, length = 20</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>property_city</td>
<td>character, length = 20</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>property_state</td>
<td>character, length = 2</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>property_zip</td>
<td>character, length = 10</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>number_units</td>
<td>character, length = 3</td>
<td>Numeric</td>
</tr>
</tbody>
</table>

* is a unique identifier.

Figure 3-2. Property Description
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Characteristics</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>property_id *</td>
<td>character, length = 6</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>purchase_price</td>
<td>numeric, length = 10</td>
<td>$9,999,999</td>
</tr>
<tr>
<td>down_payment</td>
<td>numeric, length = 8</td>
<td>$999,999</td>
</tr>
<tr>
<td>mortgage_amount</td>
<td>numeric, length = 10</td>
<td>$9,999,999</td>
</tr>
<tr>
<td>mortgage_rate</td>
<td>numeric, length = 5</td>
<td>99.99</td>
</tr>
<tr>
<td>mortgage_length</td>
<td>numeric, length = 3</td>
<td>999 (months)</td>
</tr>
<tr>
<td>closing_points</td>
<td>numeric, length = 2</td>
<td>99</td>
</tr>
<tr>
<td>loan_start_date</td>
<td>character, length = 8</td>
<td>MM/DD/YY</td>
</tr>
<tr>
<td>loan_type</td>
<td>character, length = 1</td>
<td>{fixed, ARM, balloon}</td>
</tr>
<tr>
<td>closing_date</td>
<td>character, length = 8</td>
<td>MM/DD/YY</td>
</tr>
<tr>
<td>possession_date</td>
<td>character, length = 8</td>
<td>MM/DD/YY</td>
</tr>
</tbody>
</table>

Figure 3-3. Purchase Terms
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Characteristics</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>property_id *</td>
<td>character, length = 6</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>income_type</td>
<td>character, length = 8</td>
<td>{rent, security deposit, interest, other}</td>
</tr>
<tr>
<td>date_received</td>
<td>character, length = 8</td>
<td>MM/DD/YY</td>
</tr>
<tr>
<td>amount_received</td>
<td>numeric, length = 8</td>
<td>$999,999</td>
</tr>
<tr>
<td>source</td>
<td>character, length = 20</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>comment</td>
<td>character, length = 40</td>
<td>Alphanumeric</td>
</tr>
</tbody>
</table>

Figure 3-4. Income
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Characteristics</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>property_id *</td>
<td>character, length = 6</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>expense_type</td>
<td>character, length = 18</td>
<td>{advertising, auto/travel, cleaning/maint, commissions, insurance, legal/professional, mortgage, other interest, repairs, supplies, taxes, utilities, wages/salaries, other}</td>
</tr>
<tr>
<td>date_incurred</td>
<td>character, length = 8</td>
<td>MM/DD/YY</td>
</tr>
<tr>
<td>amount_paid</td>
<td>numeric, length = 11</td>
<td>$999,999.00</td>
</tr>
<tr>
<td>payee</td>
<td>character, length = 20</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>payment_record</td>
<td>character, length = 6</td>
<td>{check, cash, credit, other}</td>
</tr>
<tr>
<td>comment</td>
<td>character, length = 40</td>
<td>Alphanumeric</td>
</tr>
</tbody>
</table>

Figure 3-5. Expenses
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Characteristics</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>property_id *</td>
<td>character, length = 6</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>owner_last_name</td>
<td>character, length = 15</td>
<td>Alphabetic</td>
</tr>
<tr>
<td>owner_first_name</td>
<td>character, length = 10</td>
<td>Alphabetic</td>
</tr>
<tr>
<td>owner_street_addr</td>
<td>character, length = 20</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>owner_city</td>
<td>character, length = 20</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>owner_state</td>
<td>character, length = 2</td>
<td>Alphabetic</td>
</tr>
<tr>
<td>owner_zip</td>
<td>character, length = 5</td>
<td>Numeric</td>
</tr>
<tr>
<td>%_of_ownership</td>
<td>numeric, length = 6</td>
<td>100.00%</td>
</tr>
</tbody>
</table>

Figure 3-6. Ownership
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Characteristics</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>property_id *</td>
<td>character, length = 6</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>tenant_last_name</td>
<td>character, length = 15</td>
<td>Alphabetic</td>
</tr>
<tr>
<td>tenant_first_name</td>
<td>character, length = 10</td>
<td>Alphabetic</td>
</tr>
<tr>
<td>tenant_street_addr</td>
<td>character, length = 20</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>tenant_city</td>
<td>character, length = 20</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>tenant_state</td>
<td>character, length = 2</td>
<td>Alphabetic</td>
</tr>
<tr>
<td>tenant_zip</td>
<td>character, length = 5</td>
<td>Numeric</td>
</tr>
<tr>
<td>unit_number</td>
<td>character, length = 3</td>
<td>Alphanumeric</td>
</tr>
</tbody>
</table>

Figure 3-7. Tenants
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Characteristics</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>property_id</td>
<td>character, length = 6</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>unit_number</td>
<td>character, length = 3</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>unit_rent</td>
<td>numeric, length = 8</td>
<td>$999,999 (\text{/mo})</td>
</tr>
<tr>
<td>beg_lease_date</td>
<td>character, length = 8</td>
<td>MM/DD/YY</td>
</tr>
<tr>
<td>end_lease_date</td>
<td>character, length = 8</td>
<td>MM/DD/YY</td>
</tr>
<tr>
<td>security_deposit</td>
<td>numeric, length = 8</td>
<td>$999,999</td>
</tr>
</tbody>
</table>

Figure 3-8. Lease Terms
Figure 3-9. E-R Relationship Diagram
1.0 Main Menu  Real Estate Investing System

1.1 Property Information
   a. Income Statement
   b. Property Description
   c. Purchase Terms
   d. Ownership Records

1.2 Rental Information
   e. Tenant Records
   f. Lease Terms

1.3 Income Information
   g. Income Records
   h. Income Reports

1.4 Expense Information
   i. Expense Records
   j. Expense Reports

Enter a-j [_] or [ESC] to quit or [F1] HELP or [F10] to enter commands directly

Figure 3-10. Main Menu
Chapter Four: Implementation of the Real Estate Investing Application

4.1 Using dBASEIII PLUS

The database package selected to use in implementing the Real Estate Investing application is dBASEIII PLUS. dBASEIII PLUS offers both a menu interface and a command language interface. The menu interface is called the Assistant, the command language interface is the "." prompt which is also used in dBASEIII. The Assistant interface was designed to be more user-friendly than the interfaces previously available in the dBASEII and dBASEIII packages.

The dBASEIII PLUS interfaces illustrate many of the features which designers are including in user-friendly software packages.

The screen format for dBASEIII PLUS is relatively fixed. If the Assistant option is chosen, the top line of the screen is reserved for a menu line. Menu choices on the top line can be selected by either positioning a highlighted bar or by typing the first letter of the option name. When the user selects one of these menu options, a more detailed pull-down menu for the user...
appears on the screen. Menu choices on the pull-down menus can only be selected by positioning the highlighted bar.

There is a status bar at the bottom of the screen that displays information on the current values of various optional settings. Below the status bar, there two more lines used by the system. These are the navigation line and the message line. These two lines display instructions and information about the currently highlighted option. There is one more line that is part of the fixed screen format. This is the action line, which appears above the status bar. This line displays the dBASEIII command which the user is building through his/her successive menu choices.

On-line HELP is available to the user any time a menu option is highlighted. The same key, [F1], is always used to request HELP.

The "'. prompt" interface allows the experienced dBASEIII PLUS user to enter commands directly.

4.2 Implementation of the Real Estate Investing System

The Real Estate Investing System is implemented in
dBASE III Plus as a menu-driven application. The menu interface is hierarchically organized. The interface is composed of 12 selection menus and 14 data screens which allow the user to add, edit, delete, and display data. The program connecting the menus and screens together and providing the investment modeling capability is written in the dBASE III Plus programming language.

A complete set of the interface menus and screens is contained in Appendix B. The source code is contained in Appendix C.

4.3 Using the Real Estate Investing System

In order to run the application, the user must have an IBM-compatible PC with 256K memory and, optionally, a printer in order to produce the hard-copy reports. The user must also have a copy of dBASE III Plus.

After the user has installed dBASE III Plus on his/her PC, the user must install the diskette containing the application program. The user's data will be written on this diskette. The user invokes the application by typing 'dbase reis'. The application program will then lead the user through the real estate investing and
record-keeping process.
Chapter Five: Summary

5.1 Conclusions

The guidelines proposed in the literature do not provide a complete model for developing a user-friendly interface. The guidelines discuss subsets of the elements which must be included in a model of a user-friendly interface. Consequently, trying to design an interface using these is a difficult job. Following the guidelines that currently exist will not guarantee that the resulting interface will be viewed as "friendly" by its users.

It will take more study and research to change the design of user-friendly interfaces from an art form to a science. While much work is being done in this area, the test population that is being used in most experiments is too small to justify extending the results to the entire user community. The human factors engineers and developers currently working in this area are often unable to duplicate other experimenters' test results.

5.2 Future Extensions

Future directions for interface design should include a
screen design language. This language would allow the designer to display screen formats, to show field limitations (such as templates for field data format, indicators to show whether the field is input or a displayed value, error checking on the domain of the field itself or in relation to values that other fields are taking), and to describe relationships between screens. A screen design language would allow for rapid-prototyping of the user interface. The interface could then be evaluated for friendliness before the system is implemented.

Future extensions of the Real Estate Investment system would involve collecting feedback about the user-interface and modifying the interface based on these responses.

In terms of functionality, the Real Estate Investment system described here is very rudimentary. Future extensions to the functionality of the system would include more sophisticated modeling techniques to provide ranges instead of one optimal value, evaluating the investment in terms of the present and future value of money, and providing the user with a complete picture of how an investment will affect his/her financial situation.
BIBLIOGRAPHY


[Coh185] Cohill, A. M.; Williges, R.C., "Retrieval of HELP Information for Novice Users of


1985.


[Poll85] Pollack, M. E., "Information Sought and Information Provided: An Empirical Study of


[Tenn83] Tennant, H.R.; Ross, K.M.; Thompson, C.W., "Usable Natural Language Interfaces Through Menu-Based Natural Language Understanding", 58


APPENDIX A - Examples

Sample Income Property Statement. The property used in this example has six units. The user thinks the seller will sell the property for $173,900. The user knows he/she can get a 30-year fixed rate mortgage at 10.25% by putting 20% down. The user is using the seller's land value, rental information, real estate taxes, and expense information for this analysis. The user could use estimates for these items if he/she thinks that the seller's figures are inaccurate. The user wants to know if this property will generate a positive cash flow. The information that the user supplies for this analysis is highlighted.

Income Property Statement

Property_street_address: 110 Boardwalk
Property_city: Monopolyville
Property_state: IL
Property_zipcode: 60546
Land Value: $30,000

1. Asking Price: $175,000
2. Purchase Price: $173,900
3. Down payment: 20%

4. Income grid
# of units @ actual rental fair rental total actual
   per month per month monthly rent
2    $625  $650  $15,000
2    $550  $550  $13,200
1    $510  $550  $6,120
1    $510  $510  $6,120

5. Annual Gross Income: $40,440
6. Is there any other income from this property? Y

source                yearly_income_generated
garage                $360
wasser/dryer          $720

7. Scheduled Gross Income: $41,520 (this is 5. + 6.)
8. Vacancy/Credit Loss Factor: 25%
9. Effective Gross Income: $31,140 (this is 7. - 8.)
10. Fixed Expenses (yearly)
a. Property Taxes $3,643
b. Insurance $689
c. Advertising $75

11. Operating Expenses (yearly)
d. Auto & Travel $225  
e. Cleaning & Maintenance $2,310  
f. Commissions $0  
g. Legal & Other  
   Professional $325  
h. Interest (on loans other than mortgage) $0  
i. Repairs $1,600  
j. Supplies $200  
k. Utilities $2,000  
l. Wages & Salaries $0  
m. Other $0  

Total Annual Expenses $11,067  

12. Annual Net Income: $20,081 (this is 9. - Total Annual Expenses)  

13. Debt Service  
   Amount of 1st Mortgage: $139,100 (rounded to nearest $100 from 20% down payment)  
   Mortgage rate: 10.25%  
   Mortgage length: 360 months  
   Yearly Principal and Interest Paid: $14,958  
   Amount of 2nd Mortgage: $0  
   (If there was a 2nd mortgage, rate, length, and yearly principal and interest would be shown)  

14. Cash Flow: $5,123 Positive! (this is 12. - 13.)  
   (If Cash Flow was negative, the Cash Flow using Fair rents would be calculated next.)  

15. Other Advantages  
   a. Annual Depreciation: $7,994 (this is using straight line depreciation over 18 years:  
      (2. - Land value )/18  
   b. Equity Build Up after 1st year: $744.85  
   c. Appreciation rate in this area: 5%  
      Appreciation after 1st year: $8,695  
      This property will then be worth: $182,595  

Examples of the types of data which can be stored in the system are  

Property Description Information  

61
property_id:  Easy
property_street_address:  115 Easy Street
property_city:  Manhattan
property_state:  KS
property_zipcode:  66506
number_of_units:  3

property_id:  IX
property_street_address:  1200 Warrenville Road
property_city:  Naperville
property_state:  IL
property_zipcode:  66506
number_of_units:  1

Ownership Information

property_id:  IX
owner_name:  Ronna Rykowski
owner_street:  1234 Any Street
owner_city:  Naperville
owner_state:  IL
owner_zip:  60032
percentage_of_ownership:  50

property_id:  IX
owner_name:  Leslie Eaton
owner_street:  115 E. George Street
owner_city:  Naperville
owner_state:  IL
owner_zip:  60032
percentage_of_ownership:  50

property_id:  Easy
owner_name:  Leslie Eaton
owner_street:  115 E. George Street
owner_city:  Naperville
owner_state:  IL
owner_zip:  60032
percentage_of_ownership:  100

Purchase Terms

property_id:  Easy
purchase_price:  $172,000
down_payment:  $10,000
mortgage_amount:  $162,000
mortgage_terms:  10.5%
mortgage_length:  360 months
closing_date:  10/10/86
possession_date: 12/15/86

Property Income Information

property_id: IX
income_type: rent
date_received: 10/12/86
amount_received: $3,600
payee: ATT
comment: October

Property Expense Information

property_id: IX
expense_type: repair
date_paid: 10/02/86
amount_paid: $12,224.50
payee: Joe's Cheap Roofing
payment_method: check
comments: replace tiles on roof, check #123
Appendix B - Menus and Screens

THE REAL ESTATE INVESTMENT SYSTEM

by

Leslie A. Eaton

Figure A - 1. Initial Screen
1.0 MAIN MENU

1.1 Property Information
   a. Income Property Statement
   b. Property Description
   c. Purchase Terms
   d. Ownership Records

1.3 Income Information
   g. Income Records
   h. Income Reports

1.2 Rental Information
   e. Tenant Records
   f. Lease Terms

1.4 Expense Information
   i. Expense Records
   j. Expense Reports

Enter a-j [_] or [F10] to quit
[F3] for HELP or [ESC] to enter DBASE commands directly
Figure A - 3. Property Identifier Selection
1.1 Property Information  a. Income statement

Property Street Address:
Property City:
Property State: Zipcode:

1. Asking Price:
2. Purchase Price:
3. Down Payment:

4. Income Grid:
   # of units  actual rental  fair rental  total actual
   per month   per month   monthly rent

5. Annual Gross Income:

PgDn - To continue to next page
1.1 Property Information a. Income statement

6. Is there any other income for this property? (y/n):

7. Schedule Gross Income (5. + 6.):

8. Vacancy/Credit Loss Factor:

9. Effective Gross Income (7. - 8.):

10. Fixed Expenses (yearly)
    a. Property taxes:
    b. Insurance:

Figure A - 5. Income Statement - Page 2
1.1 Property Information  
   a. Income statement

11. Operating Expenses (yearly)
   c. Advertising
   d. Auto & Travel
   e. Cleaning & Maintenance
   f. Commissions
   g. Legal & Professional
   h. Interest
   i. Repairs
   j. Supplies
   k. Utilities
   l. Wages & Salaries
   m. Other

   Total Annual Expenses

1.1 Property Information  a. Income statement

13. Debt Service
   Amount of 1st Mortgage: 
   Mortgage rate: 
   Mortgage length: months 
   Yearly Principal and Interest Paid:


15. Other Advantages:
   a. Annual Depreciation:
      (2. - Land Value) / 18 years
   b. Equity Build up after 1st year:
   c. Appreciation rate in this area:  
      Appreciation after 1st year:
      This Property will then be worth:

F2 - Print Income Statement
F9 - To save this data in a temporary file
F10 - To save this data in the property data base
ESC - Leave process (will not save data)
Select desired action from Function Keys

COMMANDS:

F3 - HELP
F5 - ADD Record
F7 - EDIT RECORD
F9 - DELETE RECORD

F4 - FIND A PROPERTY IDENTIFIER
F6 - DISPLAY A PROPERTY DESCRIPTION
F8 - PRINT THIS FILE
F10 - RETURN TO THE MAINMENU
1.1 Property Information  b. Property Description

PROPERTY IDENTIFIER:

STREET:
CITY: STATE: ZIPCODE:

NUMBER OF UNITS: 0

COMANDS: EDIT

CURSOR: <-- -->  CHAR: <--  WORD: Home End
UP DOWN  FIELD: cursor field Page: PgUp PgDn
DELETE  CHAR: Del  FIELD: ^Y  RECORD: ^U
Insert Mode: Ins  Exit/Save: ^End  Abort: Esc  Memo: ^Home
1.1 PROPERTY INFORMATION   c. PURCHASE TERMS

Select desired action from Function Keys

<table>
<thead>
<tr>
<th>COMMANDS:</th>
<th>BUSY</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3 - HELP</td>
<td>F4 - FIND A PROPERTY IDENTIFIER</td>
</tr>
<tr>
<td>F5 - ADD Record</td>
<td>F6 - DISPLAY A PROPERTY DESCRIPTION</td>
</tr>
<tr>
<td>F7 - EDIT RECORD</td>
<td>F8 - PRINT THIS FILE</td>
</tr>
<tr>
<td>F9 - DELETE RECORD</td>
<td>F10 - RETURN TO THE MAIN MENU</td>
</tr>
</tbody>
</table>
### PROPERTY IDENTIFIER

<table>
<thead>
<tr>
<th>Property Information</th>
<th>c. PURCHASE TERMS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PURCHASE PRICE:</strong></td>
<td>$</td>
</tr>
<tr>
<td><strong>DOWN PAYMENT:</strong></td>
<td>$</td>
</tr>
<tr>
<td><strong>AMOUNT OF MORTGAGE:</strong></td>
<td>$</td>
</tr>
<tr>
<td><strong>MORTGAGE INTEREST RATE:</strong></td>
<td>.%</td>
</tr>
<tr>
<td><strong>LAND VALUE:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>LOAN TYPE:</strong></td>
<td>F:Fixed, A:ARM, B:Balloon</td>
</tr>
<tr>
<td><strong>DATES:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>LOAN START:</strong></td>
<td>/ /</td>
</tr>
<tr>
<td><strong>CLOSING:</strong></td>
<td>/ /</td>
</tr>
<tr>
<td><strong>POSSESSION:</strong></td>
<td>/ /</td>
</tr>
<tr>
<td><strong>INTEREST POINTS PAID:</strong></td>
<td>.%</td>
</tr>
<tr>
<td><strong>LENGTH OF LOAN IN MONTHS:</strong></td>
<td>months</td>
</tr>
</tbody>
</table>

**Commands:** EDIT

<table>
<thead>
<tr>
<th>Cursor</th>
<th>Up/Down</th>
<th>Delete</th>
<th>Insert Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;-- --&gt;</td>
<td>Field: cursor cursor</td>
<td>Char: Del</td>
<td>Exit/Save: ^End</td>
</tr>
<tr>
<td>Char:</td>
<td>Field: ^Y</td>
<td>Record: ^U</td>
<td>Abort: Esc</td>
</tr>
<tr>
<td>Word: Home End</td>
<td>Page: PgUp PgDn</td>
<td>Memo: ^Home</td>
<td></td>
</tr>
</tbody>
</table>
1.1 PROPERTY INFORMATION  d. OWNERSHIP RECORDS

Select desired action from Function Keys

COMMANDS:       BUSY
F3 - HELP        F4 - FIND A PROPERTY IDENTIFIER
F5 - ADD Record  F6 - DISPLAY A PROPERTY DESCRIPTION
F7 - EDIT RECORD F8 - PRINT THIS FILE
F9 - DELETE RECORD F10 - RETURN TO THE MAINMENU
1.1 PROPERTY INFORMATION  d. OWNERSHIP

PROPERTY IDENTIFIER:

LAST NAME:  FIRST NAME:  
ADDRESS:  
CITY:  , STATE  ZIPCODE

PERCENT OF OWNERSHIP:  %

COMMANDS:  EDIT

| CURSOR:  |-- --|  UP DOWN  | DELETE  | Insert Mode:  Ins |
| Char:  <- ->  | Field:  cursor cursor  | Char:  Del  | Exit/Save:  ^End |
| Word:  Home End  | Page:  PgUp PgDn  | Field:  ^Y  | Abort:  Esc |
|            | Record:  ^U  |            | Memo:  ^Home |
Figure A - 14. Tenant Records Selection
1.2 RENTAL INFORMATION e. TENANT RECORDS

PROPERTY IDENTIFIER:

LAST NAME:                       FIRST NAME:

STREET:                           , STATE:     ZIPCODE:

CITY:                             

UNIT NUMBER:

COMMANDS: EDIT

CURSOR: <- -> Char: <- -> Word: Home End

UP DOWN Field: cursor cursor Page: PgUp PgDn

DELETE Char: Del Field: ^Y Record: ^U

Insert Mode: Ins Exit/Save: ^End Abort: Esc Memo: ^Home
Select desired action from Function Keys

**COMMANDS:**

- F3 - HELP
- F5 - ADD Record
- F7 - EDIT RECORD
- F9 - DELETE RECORD

**BUSY**

- F4 - FIND A PROPERTY IDENTIFIER
- F6 - DISPLAY A PROPERTY DESCRIPTION
- F8 - PRINT THIS FILE
- F10 - RETURN TO THE MAINMENU
1.2 RENTAL INFORMATION  f. LEASE TERMS

PROPERTY IDENTIFIER:

UNIT NUMBER:                      UNIT RENT: $
LEASE BEGINNING DATE: / /                            LEASE END DATE: / /
DEPOSIT AMOUNT: $

COMMANDS: EDIT

CURSOR: <-- -->
Char: <--
Word: Home End

UP DOWN
Field: cursor cursor
Page: PgUp PgDn

DELETE
Char: Del
Field: ^Y
Record: ^U

Insert Mode: Ins
Exit/Save: ^End
Abort: Esc
Memo: ^Home
1.3 INCOME INFORMATION  9. INCOME RECORDS

Select desired action from Function Keys

<table>
<thead>
<tr>
<th>COMMANDS:</th>
<th>BUSY</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3</td>
<td>HELP</td>
</tr>
<tr>
<td>F5</td>
<td>ADD Record</td>
</tr>
<tr>
<td>F7</td>
<td>EDIT RECORD</td>
</tr>
<tr>
<td>F9</td>
<td>DELETE RECORD</td>
</tr>
<tr>
<td>F4</td>
<td>FIND A PROPERTY IDENTIFIER</td>
</tr>
<tr>
<td>F6</td>
<td>DISPLAY A PROPERTY DESCRIPTION</td>
</tr>
<tr>
<td>F8</td>
<td>PRINT THIS FILE</td>
</tr>
<tr>
<td>F10</td>
<td>RETURN TO THE MAINMENU</td>
</tr>
</tbody>
</table>
1.3 INCOME INFORMATION  q. INCOME RECORDS

PROPERTY IDENTIFIER:

TYPE OF INCOME:   DATE RECEIVED:  //   AMOUNT RECEIVED: $   SOURCE OF INCOME:

COMMENT:

COMMANDS: EDIT

CURSOR:  <-- -->  Char:  <-- -->  Word: Home End
Field: cursor cursor  Page: PgUp PgDn

UP DOWN

DELETE
Char: Del  Field: ^Y  Record: ^U

Insert Mode: Ins  Exit/Save: ^End  Abort: Esc  Memo: ^Home
1.3 INCOME INFORMATION  h. INCOME REPORTS

Select the type of income report desired:

1. Summary for the year.
2. Summary for the year by month.
3. Complete listing for property.

Enter the report you want printed (1, 2 or 3).

F3 - HELP  F10 - Return to previous menu
Figure A - 21. Income Reports Year Selection

1.3 INCOME INFORMATION
h. INCOME REPORTS

Enter the year to be reported:

F10 - Return to previous menu
Select desired action from Function Keys

COMMANDS:

F3 - HELP
F5 - ADD Record
F7 - EDIT RECORD
F9 - DELETE RECORD

F4 - FIND A PROPERTY IDENTIFIER
F6 - DISPLAY A PROPERTY DESCRIPTION
F8 - PRINT THIS FILE
F10 - RETURN TO THE MAINMENU
PROPERTY IDENTIFIER:

EXPENSE TYPE:          DATE INCURRED:  /  /

AMOUNT PAID:          PAYEE:

PAYMENT RECORD:

COMMENT:

COMMANDS: EDIT

EXPENSE TYPES: 1: advertising  2: auto/travel  3: clean/maintenance
4: commissions  5: insurance   6: legal/profess.  7: mortgage
8: other interest 9: repairs    10: supplies   11: taxes
12: utilities   13: wages/salaries 14: other

Ctrl-End Save data and EXIT    ESC - Exit Without saving data
1.4. EXPENSE INFORMATION  j. EXPENSE REPORTS

Select the type of expense report desired:

1. Summary for the year.
2. Summary for the year by month.
3. Summary for the year by type.
4. Complete listing for property.

Enter the report you want printed (1-4).

F3 - HELP  F10 - Return to previous menu
Figure A - 26. Expense Reports Year Selection
Appendix C - Source Code

* Program AEFOOTER
* Puts footer for edit and add commands
*
@ 14, 1 SAY "COMMAN D S: "
@ 17, 2 SAY "CURSOR: " UP DOWN
DELETE Insert
Mode: Ins"
@ 18, 3 SAY "Char: " Field: cursor cursor
Char: Del Exit/Save: "End"
@ 19, 3 SAY "Word: Home End Page: PgUp PgDn
Field: "Y Abort: Esc"
@ 20, 22 SAY " Record: " ^U
Memo: "Home"
@ 15, 57 TO 21, 79 DOUBLE
@ 15, 0 TO 21, 19 DOUBLE
@ 15, 20 TO 15, 40 DOUBLE
@ 21, 20 TO 21, 40 DOUBLE
@ 15, 41 TO 21, 57 DOUBLE

91
Program: EXPENRPT.PRG

* Produces the Expense report 1.4.j

select 1
use propdesc
select 2
use expenses
@ 1, 29 SAY "EXPENSE REPORT"
@ 3, 32 SAY "PROPERTY"
@ 5, 16 SAY "STREET:"
@ 5, 28 GET PROPDESC->STREET
@ 6, 18 SAY "CITY:"
@ 6, 28 GET PROPDESC->CITY
@ 7, 17 SAY "STATE:"
@ 7, 28 GET PROPDESC->STATE
@ 8, 15 SAY "ZIPCODE:"
@ 8, 28 GET PROPDESC->ZIP
@ 11, 3 SAY "DATE EXPENSE AMOUNT PAYEE PAYMENT"
@ 12, 2 SAY "INCURRED TYPE"
@ 13, 2 GET EXPENSES->INCURRED
@ 13, 30 GET EXPENSES->AMOUNT
@ 13, 44 GET EXPENSES->PAYEE
@ 13, 67 GET EXPENSES->PAYMENT
close all
Program EXPENSES.FMT

Screen format for expense records

@ 1, 1 SAY "1.4 EXPENSE INFORMATION"
i. EXPENSE RECORDS"
@ 3, 16 SAY "PROPERTY IDENTIFIER:" 
@ 3, 38 GET EXPENSES->PROP_ID
@ 5, 5 SAY "EXPENSE TYPE:" 
@ 5, 22 GET EXPENSES->TYPE 
@ 5, 42 SAY "DATE INCURRED:" 
@ 5, 58 GET EXPENSES->INCURRED 
@ 7, 6 SAY "AMOUNT PAID:" 
@ 7, 22 GET EXPENSES->AMOUNT 
@ 7, 50 SAY "PAYEE:" 
@ 7, 58 GET EXPENSES->PAYEE FUNCTION "$I7" 
@ 10, 24 SAY "PAYMENT RECORD:" 
@ 10, 41 GET EXPENSES->PAYMENT 
@ 12, 5 SAY "COMMENT:" 
@ 12, 15 GET EXPENSES->COMMENT 
@ 14, 2 SAY "COMMANDS:" 
@ 16, 2 SAY "EXPENSE TYPES: 1: advertising 
2: auto/travel 3: clean/maintenance" 
@ 17, 2 SAY "4: commissions 5: insurance 
6: legal/profess. 7: mortgage" 
@ 18, 2 SAY "8: other interest 9: repairs 
10: supplies 11: taxes" 
@ 19, 1 SAY "12: utilities 13: wages/salaries 
14: other" 
@ 21, 5 SAY "Ctrl-End Save data and EXIT 
ESC - Exit Without saving date"
@ 2, 0 TO 13, 79 DOUBLE 
@ 15, 0 TO 22, 79 DOUBLE
* Program EXREPORT.PRG
* Produces two request screens for Expense reports
* the selection for type of processing and the year
*
clear
SELEREP = SPACE (1)
@ 1,  1 SAY "1.4. EXPENSE INFORMATION    j. EXPENSE REPORTS"
@  4, 15 SAY "Select the type of expense report desired:"
@  6, 22 SAY "1. Summary for the year."
@  7, 22 SAY "2. Summary for the year by month."
@  8, 22 SAY "3. Summary for the year by type."
@  9, 22 SAY "4. Complete listing for property."
@11, 22 SAY "_ Enter the report you want printed (1-4)."
@11, 22 GET SELEREP PICTURE "9"
@16, 14 SAY "F3 - HELP    F10 - Return to previous menu"
@  2,  0 TO 13, 79 DOUBLE
@ 14,  0 TO 18, 79 DOUBLE
read
* Request the year of Expense reports
* YEAR = SPACE (4)
clear
@ 1, 1 SAY "1.4. EXPENSE INFORMATION  j. EXPENSE REPORTS"
@ 7, 18 SAY "Enter the year to be reported: _____"
@ 7, 50 GET YEAR PICTURE "9999"
@ 16, 22 SAY "F10 - Return to previous menu"
@ 2, 0 TO 13, 79 DOUBLE
@ 14, 0 TO 19, 79 DOUBLE
read
return
Program: INCOME.FMT

* Produces the screen for income records
*

@ 1, 0 SAY "1.3 INCOME INFORMATION g. INCOME RECORDS"
@ 3, 17 SAY "PROPERTY IDENTIFIER:
@ 3, 40 GET INCOME->PROP_ID
@ 5, 7 SAY "TYPE OF INCOME:
@ 5, 26 GET INCOME->TYPE
@ 5, 44 SAY "DATE RECEIVED:
@ 5, 60 GET INCOME->RECEIVED
@ 7, 16 SAY "AMOUNT RECEIVED: $"
@ 7, 35 GET INCOME->AMOUNT PICTURE "999,999.99"
@ 9, 16 SAY "SOURCE OF INCOME:
@ 9, 35 GET INCOME->SOURCE
@ 11, 5 SAY "COMMENT:
@ 11, 14 GET INCOME->COMMENT
@ 2, 0 TO 13, 79 DOUBLE
Program: EXPENRPT.PRG

Produces the Expense report 1.4.j

select 1
use propdesc
select 2
use expenses
@ 1, 29 SAY "EXPENSE REPORT"
@ 3, 32 SAY "PROPERTY"
@ 5, 16 SAY "STREET:"
@ 5, 28 GET PROPDESC->STREET
@ 6, 18 SAY "CITY:	"
@ 6, 28 GET PROPDESC->CITY
@ 7, 17 SAY "STATE:"
@ 7, 28 GET PROPDESC->STATE
@ 8, 15 SAY "ZIPCODE:"
@ 8, 28 GET PROPDESC->ZIP
@ 11, 3 SAY "DATE	EXPENSE	AMOUNT	PAYEE	PAYMENT"
@ 12, 2 SAY "INCURRED	TYPE"
@ 13, 2 GET EXPENSES->INCURRED
@ 13, 30 GET EXPENSES->AMOUNT
@ 13, 44 GET EXPENSES->PAYEE
@ 13, 67 GET EXPENSES->PAYMENT
close all
Program EXPENSES.FMT
Screen format for expense records

@ 1, 1  SAY "1.4 EXPENSE INFORMATION
i. EXPENSE RECORDS"
@ 3, 16 SAY "PROPERTY IDENTIFIER:"
@ 3, 38 GET EXPENSES->PROP_ID
@ 5, 5  SAY "EXPENSE TYPE:"
@ 5, 22 GET EXPENSES->TYPE
@ 5, 42 SAY "DATE INCURRED:"
@ 5, 58 GET EXPENSES->INCURRED
@ 7, 6  SAY "AMOUNT PAID:"
@ 7, 22 GET EXPENSES->AMOUNT
@ 7, 50 SAY "PAYEE:"
@ 7, 58 GET EXPENSES->PAYEE FUNCTION "S17"
@ 10, 24 SAY "PAYMENT RECORD:"
@ 10, 41 GET EXPENSES->PAYMENT
@ 12, 5  SAY "COMMENT:"
@ 12, 15 GET EXPENSES->COMMENT
@ 14, 2  SAY "COMMANDS:"
@ 16, 2 SAY "EXPENSE TYPES:  1: advertising
2: auto/travel   3:
clean/maintenance"
@ 17, 2 SAY "4: commissions   5: insurance
6: legal/profess.
7: mortgage"
@ 18, 2 SAY "8: other interest 9: repairs
10: supplies   11: taxes"
@ 19, 1 SAY "12: utilities   13: wages/salaries
14: other"
@ 21, 5 SAY "Ctrl-End Save data and EXIT
ESC - Exit Without saving data"
@ 2, 0 TO 13, 79 DOUBLE
@ 15, 0 TO 22, 79 DOUBLE
Program EXREPORT.PRG
* Produces two request screens for Expense reports
* the selection for type of processing and the year
*
clear
SELEREP = SPACE (1)
@ 1, 1 SAY "1.4. EXPENSE INFORMATION"
j. EXPENSE REPORTS"
@ 4, 15 SAY "Select the type of expense report desired:"
@ 6, 22 SAY "1. Summary for the year."
@ 7, 22 SAY "2. Summary for the year by month."
@ 8, 22 SAY "3. Summary for the year by type."
@ 9, 22 SAY "4. Complete listing for property."
@ 11, 22 SAY "_. Enter the report you want"
printed (1-4)."
@ 11, 22 GET SELEREP PICTURE "9"
@ 16, 14 SAY "F3 - HELP"
F10 - Return to previous menu"
@ 2, 0 TO 13, 79 DOUBLE
@ 14, 0 TO 18, 79 DOUBLE
read
* Request the year of Expense reports *

YEAR = SPACE (4)
clear
@ 1, 1 SAY "1.4. EXPENSE INFORMATION"
j. EXPENSE REPORTS"
@ 7, 18 SAY "Enter the year to be reported: ____"
@ 7, 50 GET YEAR PICTURE "9999"
@ 16, 22 SAY "F10 - Return to previous menu"
@ 2, 0 TO 13, 79 DOUBLE
@ 14, 0 TO 19, 79 DOUBLE
read
return
Program: INCOME.FMT
* Produces the screen for income records
*
@ 1, 0 SAY "1.3 INCOME INFORMATION  g. INCOME RECORDS"
@ 3, 17 SAY "PROPERTY IDENTIFIER:"
@ 3, 40 GET INCOME->PROP_ID
@ 5, 7 SAY "TYPE OF INCOME:"
@ 5, 26 GET INCOME->TYPE
@ 5, 44 SAY "DATE RECEIVED:"
@ 5, 60 GET INCOME->RECEIVED
@ 7, 16 SAY "AMOUNT RECEIVED: $"
@ 7, 35 GET INCOME->AMOUNT PICTURE "999,999.99"
@ 9, 16 SAY "SOURCE OF INCOME:"
@ 9, 35 GET INCOME->SOURCE
@ 11, 5 SAY "COMMENT:"
@ 11, 14 GET INCOME->COMMENT
@ 2, 0 TO 13, 79 DOUBLE
Program: INCOME.PRG

* Provides for accepting income data from the user
*
select 1
use income
set format to income
append
@ 5, 8 say "This is an overlay"
@ 6, 8 to 10, 20 DOUBLE
return
Program: INCOMRPT.PRG

* Produces the income report
*
select 1
use propdesc
select 2
use income
@  1,  26 SAY "INCOME REPORT"
@  4,  29 SAY "PROPERTY"
@  6,  14 SAY "STREET:"
@  6,  26 GET PROPDESC->STREET
@  7,  16 SAY "CITY:"
@  7,  26 GET PROPDESC->CITY
@  8,  15 SAY "STATE:"
@  8,  26 GET PROPDESC->STATE
@  9,  13 SAY "ZIPCODE:"
@  9,  26 GET PROPDESC->ZIP PICTURE "XXXXXXXXXXXX"
@ 13,  6 SAY "DATE INCOME AMOUNT SOURCE"
@ 14,  4 SAY "RECEIVED TYPE RECEIVED"
@ 15,  4 GET INCOME->RECEIVED
@ 15,  16 GET INCOME->TYPE
@ 15,  29 SAY "$"
@ 15,  30 GET INCOME->AMOUNT
@ 15,  48 GET INCOME->SOURCE
close all
Program: INCOMS.PRG
Accepts data for creating the income statement

clear
@ 1, 1 SAY "1.1 Property Information
a. Income statement"
@ 3, 6 SAY "Property Street Address:"  
@ 4, 6 SAY "Property City:"  
@ 5, 6 SAY "Property State:  Zipcode:"  
@ 7, 2 SAY "1. Asking Price:"  
@ 8, 2 SAY "2. Purchase Price:"  
@ 9, 2 SAY "3. Down Payment:"  
@ 11, 2 SAY "4. Income Grid:"  
@ 12, 6 SAY "# of units  actual rental  fair rental  total actual"
@ 13, 21 SAY "per month  per month  monthly rent"
@ 16, 2 SAY "5. Annual Gross Income:"  
@ 20, 23 SAY "PgDn - To continue to next page"  
@ 2, 0 TO 2, 79 DOUBLE  
@ 19, 0 TO 21, 79 DOUBLE
READ

clear
@ 1, 1 SAY "1.1 Property Information
a. Income statement"
@ 4, 2 SAY "6. Is there any other income for this property? (y/n):"
@ 6, 9 SAY "Yearly income generated:"  
@ 8, 2 SAY "7. Schedule Gross Income (5. + 6.):"
@ 10, 2 SAY "8. Vacancy/Credit Loss Factor: %"
@ 12, 2 SAY "9. Effective Gross Income (7. - 8.):"
@ 14, 1 SAY "10. Fixed Expenses (yearly)"
@ 15, 5 SAY "a. Property taxes:"
@ 16, 5 SAY "b: Insurance:"
@ 20, 23 SAY "PgDn - To continue to next page"  
@ 2, 0 TO 2, 79 DOUBLE  
@ 19, 0 TO 21, 79 DOUBLE
READ

clear
@ 1, 1 SAY "1.1 Property Information
a. Income statement"
@ 3, 2 SAY "11. Operating Expenses (yearly)"
@ 4, 6 SAY "c. Advertising"
@ 5, 6 SAY "d. Auto & Travel"
@ 6, 6 SAY "e. Cleaning & Maintenance"
@ 7, 6 SAY "f. Commissions"
@ 8, 6 SAY "g. Legal & Professional"
@ 9, 6 SAY "h. Interest"
@ 10, 6 SAY "i. Repairs"
@ 11, 6 SAY "j. Supplies"
@ 12, 6 SAY "k. Utilities"

104
1. Wages & Salaries

m. Other

Total Annual Expenses

Annual Net Income (9. - Total Annual Expenses):

PgDn - To continue to next page

TO 2, 79 DOUBLE

READ clear

Property Information

Income statement

13. Debt Service

Amount of 1st Mortgage:

Mortgage rate:

Mortgage length: months

Yearly Principal and Interest Paid:


15. Other Advantages:

a. Annual Depreciation:

(2. - Land Value) / 18 years

b. Equity Build up after 1st year:

c. Appreciation rate in this area: %

Appreciation after 1st year:

This Property will then be worth:

Print Income Statement

To save this data in a temporary file

To save this data in the property database

ESC - Leave process (will not save data)

TO 2, 79 DOUBLE

TO 24, 78 DOUBLE

read clear
Program: INREPORT.PRG

Produce the income report

CLEAR
SELREP = SPACE (1)
@ 1, 1 SAY "1.3 INCOME INFORMATION h. INCOME REPORTS"
@ 4, 15 SAY "Select the type of income report desired:"
@ 6, 22 SAY "1. Summary for the year."
@ 7, 22 SAY "2. Summary for the year by month."
@ 8, 22 SAY "3. Complete listing for property."
@ 10, 22 SAY "- Enter the report you want printed
(1, 2 or 3)."
@ 10, 22 GET SELREP PICTURE "9"
@ 16, 14 SAY "F3 - HELP"
F10 - Return to previous menu"
@ 2, 0 TO 13, 79 DOUBLE
@ 14, 0 TO 18, 79 DOUBLE
read
* Request the year for Income and Expense reports
*
YEAR = SPACE (4)
clear
@ 1, 1 SAY "1.3 INCOME INFORMATION h. INCOME REPORTS"
@ 7, 18 SAY "Enter the year to be reported: ____"
@ 7, 50 GET YEAR PICTURE "9999"
@ 16, 22 SAY "F10 - Return to previous menu"
@ 2, 0 TO 13, 79 DOUBLE
@ 14, 0 TO 19, 79 DOUBLE
read
return
Program: INTRO.FMT

Displays the System introduction screen

@ 4, 18 SAY "THE REAL ESTATE INVESTMENT SYSTEM"
@ 5, 38 SAY ". . . ."
@ 6, 36 SAY ". . ."
@ 8, 33 SAY "$"
@ 9, 29 SAY "$________$"
@ 10, 26 SAY "$________________$"
@ 11, 22 SAY "$________________$"
@ 12, 26 SAY "___ _________ ___"
@ 13, 26 SAY "___ __ ________"
@ 14, 33 SAY ". . ."
@ 15, 26 SAY "___ _________"
@ 17, 32 SAY "by"
@ 19, 26 SAY "LESLIE A. EATON"
@ 11, 25 TO 15, 25 DOUBLE
@ 11, 41 TO 15, 41 DOUBLE
@ 13, 32 TO 15, 34 DOUBLE
@ 12, 28 TO 14, 30
@ 12, 36 TO 14, 38
@ 7, 35 TO 8, 36 DOUBLE
Program: LEASE.FMT

Screen format for acceptance lease data

@ 1, 0 SAY "1.2 RENTAL INFORMATION  f. LEASE TERMS"
@ 4, 17 SAY "PROPERTY IDENTIFIER:"
@ 4, 43 GET LEASE->PROP_ID
@ 7, 13 SAY "UNIT NUMBER:"
@ 7, 28 GET LEASE->UNIT_NO
@ 7, 48 SAY "UNIT RENT: $"
@ 7, 62 GET LEASE->UNIT_RENT
@ 9, 4 SAY "LEASE BEGINNING DATE:"
@ 9, 28 GET LEASE->LEASE_BEG
@ 9, 44 SAY "LEASE END DATE:"
@ 9, 62 GET LEASE->LEASE_BEG
@ 11, 10 SAY "DEPOSIT AMOUNT: $"
@ 11, 28 GET LEASE->DEPOSIT
@ 2, 0 TO 13, 79 DOUBLE
* Program: LISTPID.PRG
* List Property Identifiers for selection
* used when proper identifier is not known
*
set talk off
use propdesc index propdesc
clear
VAR1 = ' '  
DO WHILE .NOT. EOF()
    @ 1, 18 SAY "SELECT A PROPERTY IDENTIFIER"
    @ 4, 15 SAY "Property Identifier:  "
* 4, 37 GET  PROPDESC->PROP_ID
?? PROPDESC->PROP_ID
    @ 6, 20 SAY "Street Address:  "
* 6, 37 GET  PROPDESC->STREET
?? PROPDESC->STREET
    @ 9, 15 SAY "If this is the desired Property
Identifier type y"
    @ 11, 15 SAY "Otherwise hit return for the next
record: "
    @ 11, 61 GET  VAR1
    @ 17, 19 SAY "F10 - Return to previous menu"
    @ 2,  0 TO 13, 79  DOUBLE
    @ 15,  0 TO 19, 79  DOUBLE
read
IF UPPER(VAR1) = "Y"
    EXIT
ENDIF
SKIP
ENDDO
RETURN
* Program: MAINMENU.PRG
* This is the mainmenu for the real estate investment
* system. All actions come through here. The user makes
* a selection as to the type of processing in the system
* from the menu. Procedure calls are made to the programs
* for the requested function.
*
PUBLIC propid
CHOICE = SPACE(1)
* establish a loop to test all pgm
DO WHILE CHOICE <> "x"
CLEAR
do mainmenu.fmt
@ 18, 21 GET CHOICE PICTURE 'A'
READ
* test only - clear the 21st line for repeated messages
@ 23,1 SAY " 
@ 23,1 SAY "RUN:" DO CASE
    CASE CHOICE = "a"
        @ 23,6 SAY "INCOME STATEMENT"
    CASE CHOICE = "b"
        @ 23,6 SAY "PROPERTY DESCRIPTION"
        SET PROCEDURE TO PROPIINFO
        CLOSE PROCEDURE
    CASE CHOICE = "c"
        @ 23,6 SAY "PURCHASE TERMS"
    CASE CHOICE = "d"
        @ 23,6 SAY "OWNERSHIP RECORDS"
    CASE CHOICE = "e"
        @ 23,6 SAY "RENTAL INFORMATION - TENANT RECORDS"
    OTHERWISE
        @ 23,1 SAY " INCORRECT ANSWER PLEASE REENTER"
ENDCASE
WAIT "type an x to stop loop or the space bar to continue" TO CHOICE
ENDDO
RETURN
/* Program: OWNERSHI.FMT
   * Screen format for accepting owner information
   *
   @ 1, 0 SAY "1.1 PROPERTY INFORMATION  d. OWNERSHIP"
   @ 3, 12 SAY "PROPERTY IDENTIFIER:"
   @ 3, 36 GET OWNERSHI->PROP_ID
   @ 6, 5 SAY "LAST NAME:"
   @ 6, 18 GET OWNERSHI->LASTNAME
   @ 6, 41 SAY "FIRST NAME:"
   @ 6, 56 GET OWNERSHI->FIRSTNAME
   @ 8, 5 SAY "ADDRESS:"
   @ 8, 18 GET OWNERSHI->ADDRESS
   @ 10, 5 SAY "CITY:"
   @ 10, 18 GET OWNERSHI->CITY
   @ 10, 39 SAY ", STATE"
   @ 10, 49 GET OWNERSHI->STATE
   @ 10, 54 SAY "ZIPCODE"
   @ 10, 64 GET OWNERSHI->ZIP PICTURE "NNNNN NNNN"
   @ 12, 18 SAY "PERCENT OF OWNERSHIP:"
   @ 12, 41 GET OWNERSHI->PERCENTOWN
   @ 12, 48 SAY "/%"
   @ 2, 0 TO 13, 79 DOUBLE

112
* Program: PROPDESC.FMT
* Screen for accepting property description information
*
@ 1, 0 SAY "1.1 Property Information
b. Property Description"
@ 3, 15 SAY "PROPERTY IDENTIFIER:" 
@ 3, 40 GET PROPDESC->PROP_ID
@ 6, 6 SAY "STREET:" 
@ 6, 16 GET PROPDESC->STREET
@ 8, 8 SAY "CITY:" 
@ 8, 16 GET PROPDESC->CITY
@ 8, 40 SAY "STATE:" 
@ 8, 48 GET PROPDESC->STATE
@ 8, 53 SAY "ZIPCODE:" 
@ 8, 63 GET PROPDESC->ZIP PICTURE "NNNNN NNNN"
@ 11, 6 SAY "NUMBER OF UNITS:" 
@ 11, 24 GET PROPDESC->NUM_UNITS
@ 2, 0 TO 13, 79 DOUBLE
Program: PURCHASE.FMT

Screen to accept purchase information

@ 1, 0 SAY "1.1 PROPERTY INFORMATION"
@ 3, 20 SAY "PROPERTY IDENTIFIER"
@ 3, 43 GET PURCHASE->PROP_ID
@ 5, 3 SAY "PURCHASE PRICE: $"
@ 5, 21 GET PURCHASE->PRICE PICTURE "99,999,999"
@ 5, 44 SAY "AMOUNT OF MORTGAGE: $"
@ 5, 68 GET PURCHASE->MORT_AMT PICTURE "99,999,999"
@ 6, 5 SAY "DOWN PAYMENT: $"
@ 6, 21 GET PURCHASE->DOWN_PAYMT PICTURE "999,999"
@ 6, 40 SAY "MORTGAGE INTEREST RATE:"
@ 6, 68 GET PURCHASE->MORT_RATE
@ 6, 74 SAY "%"
@ 7, 52 SAY "LAND VALUE:
@ 7, 68 GET PURCHASE->LAND_VALUE
@ 8, 4 SAY "LOAN TYPE:"
@ 8, 17 GET PURCHASE->LOAN_TYPE
@ 8, 20 SAY "F:Fixed, A:ARM, B:Balloon"
@ 9, 18 SAY "DATES:"
@ 10, 3 SAY "LOAN START:"
@ 10, 17 GET PURCHASE->LOAN_START
@ 10, 38 SAY "INTEREST POINTS PAID:"
@ 10, 62 GET PURCHASE->POINTS
@ 10, 67 SAY "%"
@ 11, 6 SAY "CLOSING:"
@ 11, 17 GET PURCHASE->CLOSING
@ 12, 3 SAY "POSSESSION:"
@ 12, 17 GET PURCHASE->POSSESSION
@ 12, 38 SAY "LENGTH OF LOAN IN MONTHS:"
@ 12, 64 GET PURCHASE->MORT_LEN_M
@ 12, 68 SAY "months"
@ 2, 0 TO 13, 79 DOUBLE
* Program: REQUEST.PRG
* Request action for processing after main menu selection
* used by all programs that build the data bases
*
@  7, 18 SAY "Select desired action from Function Keys"
@  2,  0 TO 13,  79 DOUBLE
@ 14,  1 SAY "COMMANDS:"
@ 14,  72 SAY "BUSY"
@ 16,  7 SAY "F3 - HELP"
F4 - FIND A PROPERTY IDENTIFIER"
@ 17,  7 SAY "F5 - ADD Record"
F6 - DISPLAY A PROPERTY
DESCRIPTION"
@ 18,  7 SAY "F7 - EDIT RECORD"
F8 - PRINT THIS FILE"
@ 19,  7 SAY "F9 - DELETE RECORD"
F10 - RETURN TO THE MAINMENU"
@ 15,  0 TO 21,  79 DOUBLE
return
Program: TENANTS_FMT

Screen format for accepting tenant input data

@ 1, 0 SAY "1.2 RENTAL INFORMATION e. TENANT RECORDS"
@ 3, 15 SAY "PROPERTY IDENTIFIER:"
@ 3, 37 GET TENANTS->PROP_ID
@ 5, 3 SAY "LAST NAME:"
@ 5, 14 GET TENANTS->LASTNAME
@ 5, 40 SAY "FIRST NAME:"
@ 5, 54 GET TENANTS->FIRSTNAME
@ 7, 6 SAY "STREET:"
@ 7, 14 GET TENANTS->STREET
@ 9, 8 SAY "CITY:"
@ 9, 14 GET TENANTS->CITY
@ 9, 34 SAY ", STATE:"
@ 9, 43 GET TENANTS->STATE
@ 9, 48 SAY "ZIPCODE:"
@ 9, 57 GET TENANTS->ZIP PICTURE "NNNNN NNNN"
@ 11, 23 SAY "UNIT NUMBER:"
@ 11, 37 GET TENANTS->UNIT_NO
@ 2, 0 TO 13, 79 DOUBLE
DESIGN CONSIDERATIONS IN THE DEVELOPMENT OF USER-FRIENDLY INTERFACES

by

LESLIE A. EATON

B.S., University of Illinois, 1978

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas
1986
The quality of the human-computer interface is a main determinant in the success and degree of use of an interactive computer system. A new class, non-professional computer users, motivates the study and development of user-friendly interfaces to computer systems.

This paper looks at some human factors that effect the user's perception of user-friendliness. It then examines existing user interface techniques and the areas that interface designers should consider when developing user-friendly interfaces.

The requirements and design of a real estate investing application for a personal computer are used as an example environment in which to frame this study. The user-interface to this application will illustrate the techniques and design concerns described.