Requirements for an Electronic Rolodex Tool

Version 1.0
July 2000

Contents

1. Introduction

This document defines requirements for an electronic Rolodex Tool. The tool is used to store and retrieve personal information records. Each record is in a simple format containing a person's name, address, and other information.

In addition to providing useful functionality, the Rolodex Tool serves as a pedagogical example of software engineering concepts and principles. The software development artifacts for the Rolodex tool are considered part of the overall Rolodex Tool package.

1.1. Problem Statement

The functional problem to be solved by the Rolodex Tool is straightforward -- to manage a collection of information records efficiently and reliably. This problem is sufficiently simple to be of little technical interest on its own. Hence, the problem of primary interest here is pedagogical. The specific problem is to provide a small yet complete example of the software development life cycle. The example needs to be small enough to be presented in a concise fashion. At the same time, the example system must embody typical software functionality, so that students can adapt concepts illustrated in the example in their work on class software projects.

1.2. System Personnel

The personnel involved in the Rolodex Tool project are organized into the following groups:

a. system developer
b. software engineering students
c. end users
d. outside parties

The system developer is Gene Fisher. He has developed the Rolodex Tool for use by undergraduate software engineering students at Cal Poly university.

End users of the Rolodex Tool are those people who may use the tool for its actual functional purpose. As noted in the previous section, the Rolodex Tool has little functional interest as an operational piece of software. Hence, the non-academic end user community is likely to be small.

The Rolodex Tool and its development artifacts are available as public domain software for use by outside parties. The project directory is located at

`http://www.csc.calpoly.edu/˜gfisher/projects/rolodex`.

1.3. Operational Setting

There are two operational settings for the Rolodex Tool: (1) normal use as a functioning software system; (2) use as an example in software engineering courses.

Since the Rolodex Tool is intended to be as general public domain software, there is no specific operational setting in which it is intended to be installed. The setting for which it is appropriately suited is an individual or organizational environment where users maintain personal information records.

For use as a pedagogical example, the Rolodex Tool is intended to fit the curriculum used by Gene Fisher in undergraduate and graduate software engineering courses at Cal Poly University. These courses are two-quarter sequences that cover standard aspects of software engineering, with emphasis on the practical application of formal methods. For the most part, the concepts covered in these courses, and hence the concepts embodied in the Rolodex Tool, are mainstream software engineering. Other instructors may therefore find the Rolodex Tool and its development artifacts useful as course examples.

1.4. Impacts

The positive potential impacts of the Rolodex Tool as a functioning system are increased convenience and efficiency in managing personal information records. The positive impacts of the Rolodex Tool as a course example are

a. the presentation of a small, self-contained software system that students can use as a guide for their own software development work;

b. an illustration of how formal methods of can be put to practical use.

Potential negative impacts include those common to any user-oriented software system. Viz., if the system is poorly designed and implemented, it can be inconvenient to use and decrease rather than increase user productivity. If a user relies on a flawed implementation of the system to store vital information, the user may suffer inconvenience or more serious disruption of work activities if the system looses or corrupts information.

As a course example, the use of the Rolodex Tool has no significant negative impacts, unless its development methodology is considered weak or unrelated to the concepts being taught in a particular course. Such negative impacts can be easily avoided if instructors carefully examine the example before using it in a particular software engineering curriculum.

1.5. Related Systems

There are many widely used commercial systems that provide functionality comparable to the Rolodex Tool. The functionality of the Rolodex Tool is generally too limited to be offered in a stand-alone commercial tool. Rather, a variety of tools provide Rolodex-like functionality as a (small) subset of other features. In particular, any tool that allows the user easily to create, store, and locate database records provides essentially all of the functionality available in the Rolodex Tool. Popular tools that provide such functionality include Microsoft Access (http://www.microsoft.com/office/access) and AppleWorks for the MacIntosh (http://www.apple.com/appleworks).

2. Functional Requirements

The Rolodex Tool provides functions to maintain a collection of personal information cards. The tool is an electronic version of the familiar desk-top Rolodex, in which small cards are stored for easy access to peoples' names and addresses. In the electronic Rolodex, the user may add, delete, change, and find stored cards.

Following an overview of the Rolodex user interface, details of system use are presented in the following scenarios:

- adding cards to the Rolodex
- finding cards
- changing cards
- deleting cards
- file and edit details
- help commands
- data entry details
- error conditions

After the usage scenarios is a section that summarizes graphical user interface details relevant to implementations in specific operating environments.

2.1. User Interface Overview

When the user initially invokes the Rolodex tool, the screen appears as shown in Figure 1. The banner at the top of the display window identifies the Rolodex Tool and indicates that there is no currently open rolodex file. Below the banner is the command menu bar, with menus for File, Edit, Rolodex, and Help commands. Below the menu bar is the rolodex data display area, which is blank in the initial display. When the user chooses commands from the `Rolodex` menu, the data area displays appropriate information for the chosen command. Details of the data displays are covered in the scenarios starting in Section

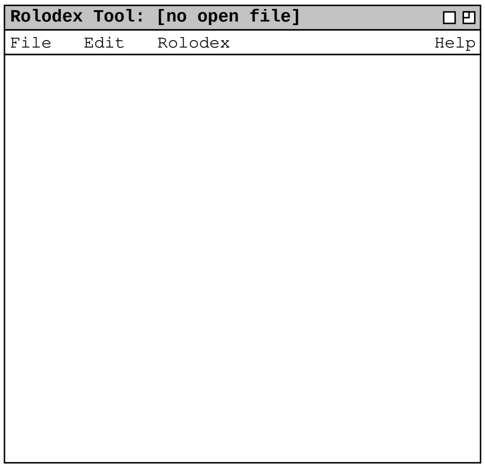

Figure 1: Initial screen.

2.2 .

2.1.1. Command Summary

Figure 2 shows an expansion of the command menus. The `File` menu contains typical commands for manipulating data files and performing other system-level functions. `File->New` opens a new, empty rolodex file. `File->Open` opens an existing rolodex from a previously saved file. `File->Save` saves the currently active rolodex on the file from which it was opened. `File->Save As` allows the current rolodex to be saved on a different file from which it was opened or most recently saved upon. `File->Print` prints the complete contents of the currently open rolodex. `File->Exit` exits the Rolodex tool, offering to save the rolodex if necessary.

The `Edit` menu contains commands for manipulating rolodex data during editing. `Edit->Undo` undoes the most recently completed `Rolodex` command. `Edit->Cut` removes and copies selected text in the current display. `Edit->Copy` copies the currently selected text without removing it. `Edit->Paste` inserts the most recently cut or copied text at the currently selected edit point.

The `Rolodex` menu contains the main processing commands of the tool. The `Rolodex->Add` command adds a new card to the currently open rolodex. The `Rolodex->Find` command finds cards in the rolodex by name. The `Rolodex->Change` and `Rolodex->Delete` commands change and delete an existing card, respectively.

The `Help` menu contains commands that present documentation for the Rolodex Tool. `Help->About` displays a brief description of the tool, including information about the tool developers and how they can be contacted. `Help->Show Quick Help` activates brief help messages that appear when the user moves the mouse over various areas of the display screen. `Help->Detailed Help` displays an online version of the complete Rolodex Tool users manual.

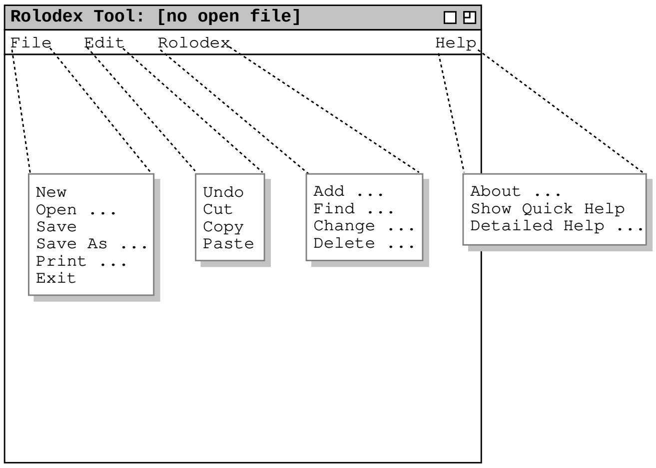

Figure 2: Command menus expanded.

2.1.2. Screen Map

Figure 3 shows a hierarchical map of the user major interface screens presented in the Rolodex functional requirements. Each image is a reduced-size thumbnail of the actual screen. In the electronic version of these requirements, the reader may click on a thumbnail image to go to the section of the requirements where the full-size screen is shown and described. The arrows between the screens indicate a transition from one screen to another, initiated by a user or system action. The italic text above each transition arrow summarizes the action that causes the transition to take place.

Table 1 shows the screen map information in textual form. The table refers to the section of the requirements where each screen is described and the figure(s) that show the full-size view of each screen.

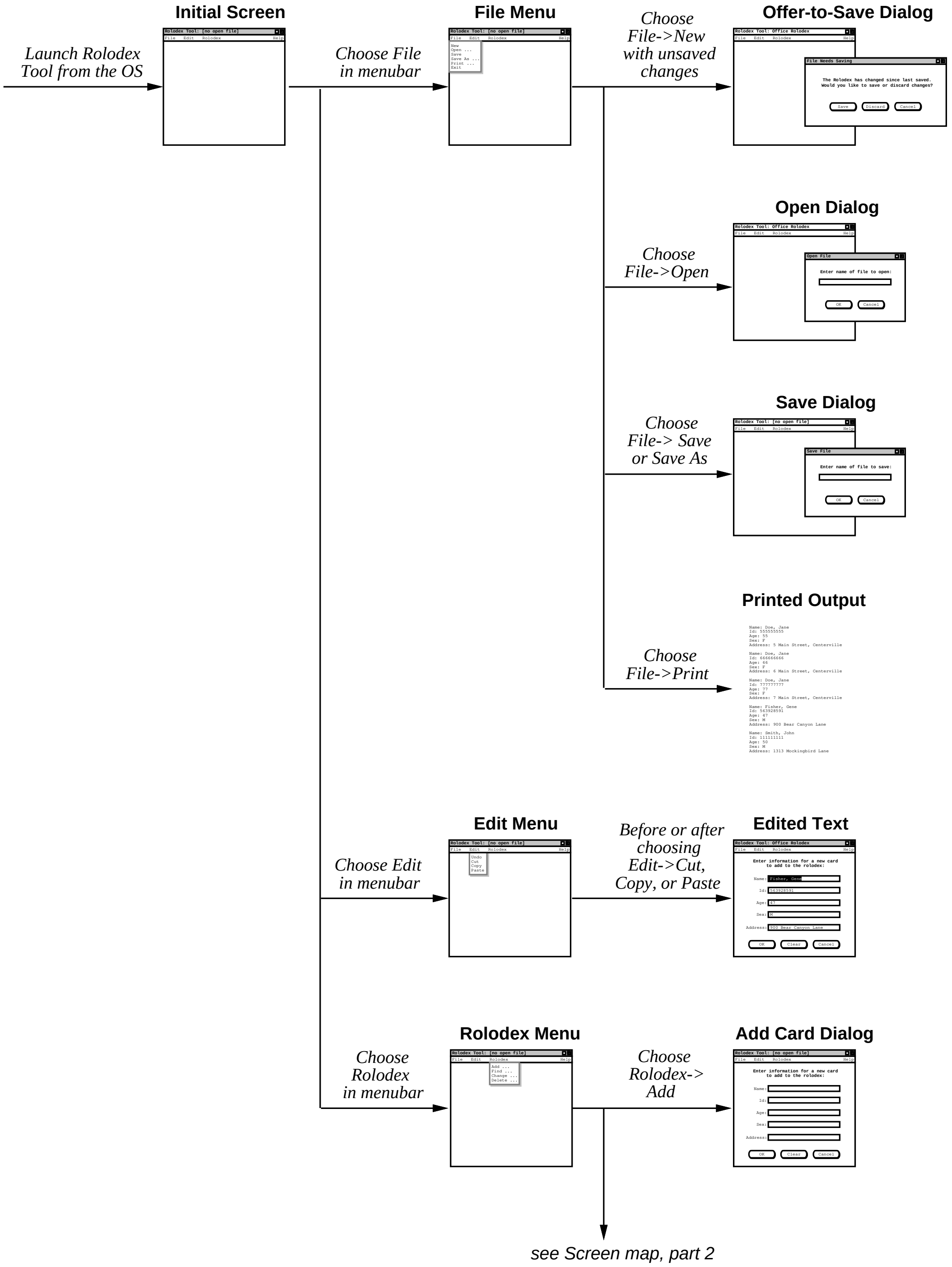

Figure 3: Screen map, part 1.

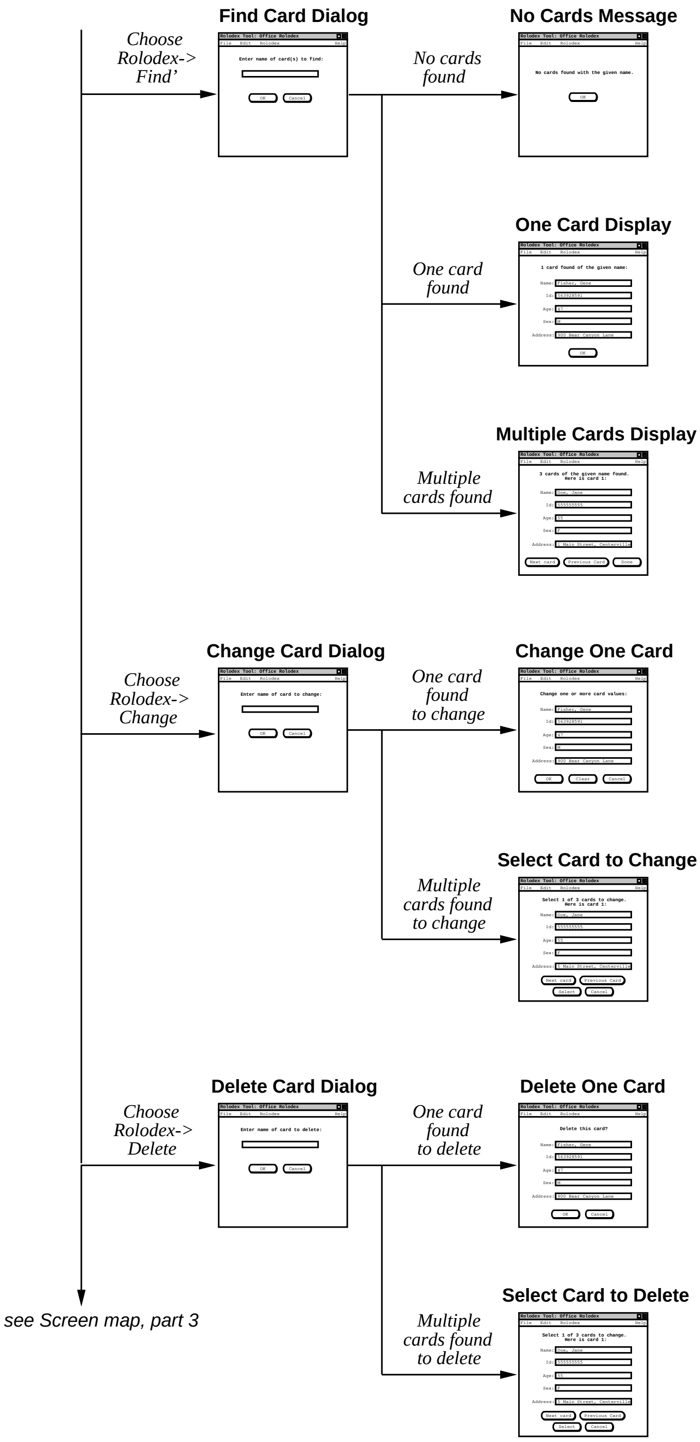

Figure 3: Screen map, part 2.

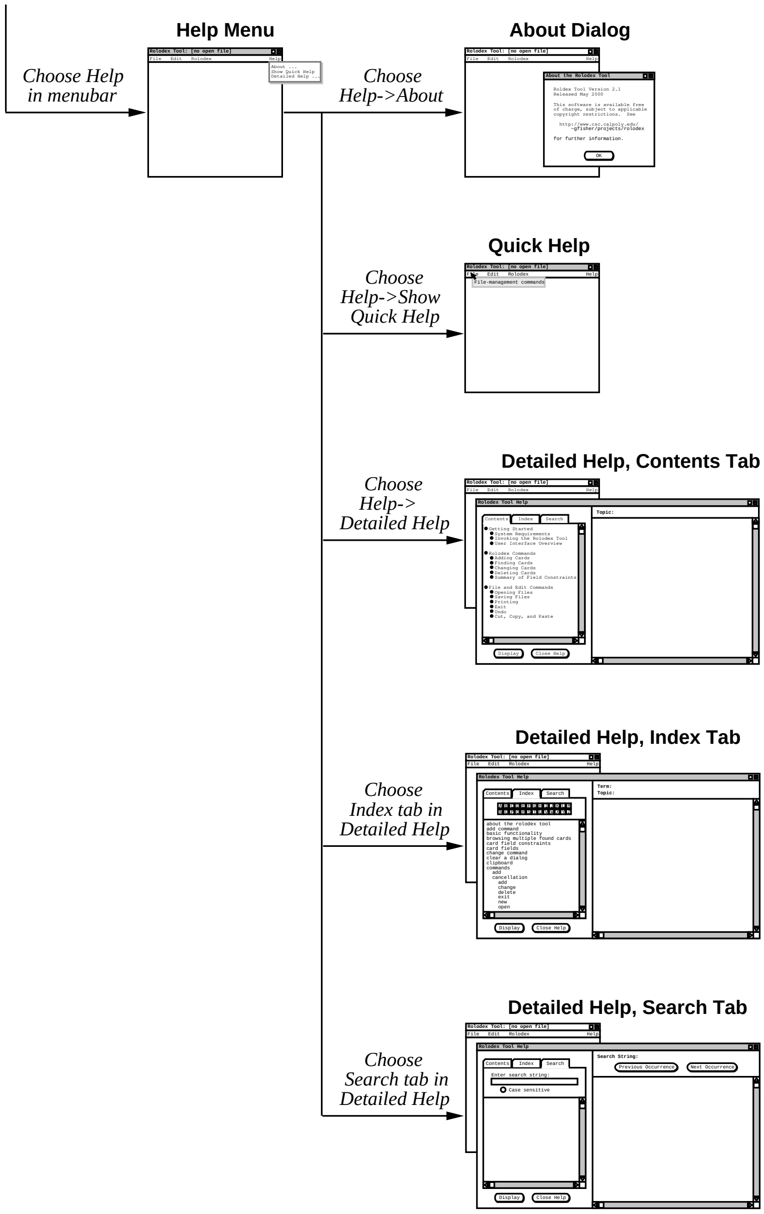

Figure 3: Screen map, part 3.

Screen Description	Associated Action	Section	Figure(s)
Initial Screen	Launch from the OS	2.1	1
File Menu	Choose `File` in menubar	2.1.1	2
Offer-to-Save Dialog	Choose `File->New` with unsaved changes	2.6.2	27
Open Dialog	Choose `File->Open`	2.6.2	29 , 30
Save Dialog	Choose `File->Save` or `Save As`	2.6.3	6 , 32
Printed Output	Choose `File->Print`	2.6.4	34
Edit Menu	Choose `Edit` in menubar	2.1.1	2
Edited Text	Before or after choosing `Edit->Cut`, `Copy`, or `Paste`	2.6.7	35 , 36 , 37
Rolodex Menu	Choose `Rolodex` in menubar	2.1.1	2
Add Dialog	Choose `Rolodex->Add`	2.2	4 , 5
Find Dialog	Choose `Rolodex->Find`	2.3	8 , 9 , 11
No Cards Message	No cards found	2.3	10
One Card Display	One card found	2.3	12
Multiple Cards Display	Multiple cards found	2.3	13
Change Dialog	Choose `Rolodex->Change`	2.4	14 , 15 , 18
One card found to change	Change one card	2.4	16 , 17
Multiple cards found to change	Select card to change	2.4	19
Delete Dialog	Choose `Rolodex->Delete`	2.5	21 , 22 , 24
One card found to delete	Delete one card	2.5	23
Multiple cards found to delete	Select card to delete	2.5	25
Help Menu	Choose `Help` in menubar	2.1.1	2
About Dialog	Choose `Help->About`	2.7.1	38
Quick Help	Choose `Help->Show Quick Help`	2.7.2	39 , 40 , 41
Detailed Help, Contents Tab	Choose `Help->Show Details`	2.7.3.1	43 , 44
Detailed Help, Index Tab	Choose `Index` tab in Detailed Help	2.7.3.2	45 , 46
Detailed Help, Search Tab	Choose `Search` tab in Detailed Help	2.7.3.3	47 , 48

Table 1: Screen map details.

2.2. Adding Cards

When the user selects the `Add` command from the `Rolodex` menu, the system updates the display area as shown in Figure 4. To add a new card, the user fills in information for each of the five fields in the add-card dialog. For example, Figure 5 shows the result of the user having typed in information for a new card. To confirm the addition of the card into the rolodex, the user presses the `OK` button at the bottom of the dialog. When the user presses `OK`, the new card is added into the Rolodex, but the display does not change. That is, the typed-in information stays in place. This allows the user to add successive cards that may have similar field values in some positions. The successive adds may be made without reselecting the `Add` command from the `Rolodex` menu. For each successive card to be added, the user edits the desired data entry fields and presses the `OK` button when the edits are complete. Each successive press of `OK` adds another new card.

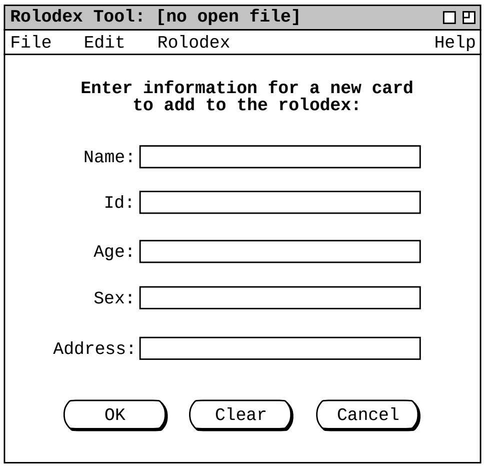

Figure 4: Add card dialog.

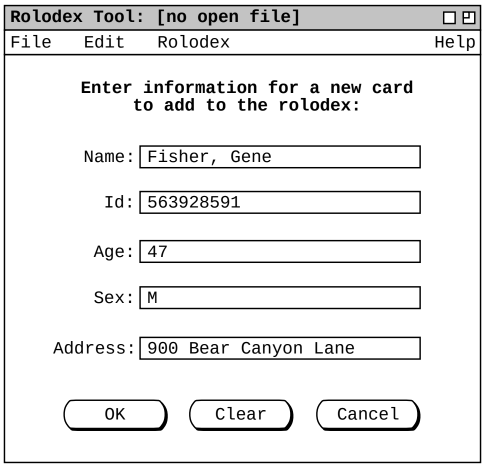

Figure 5: Add card dialog filled in.

To clear all of the typed information in the text edit boxes, the user presses the `Clear` button. This puts the add-card dialog in its initial state, as shown in Figure 4. To cancel the `Add` command entirely, the user presses `Cancel`. When `Cancel` is pressed, the entire data area is cleared, leaving it empty as in the initial rolodex display shown in Figure 1.

The system disallows two or more cards to be added with the same value of Id field. Aside from this duplication restriction, any of the other fields may be the same on two or more cards. For example, two cards may have the same name and age, as long as the Id fields differ. The Rolodex Tool enforces data-entry constraints on the values typed in for the card fields, such as the length of field values. These constraints are covered in Section 2.8 on data entry details and Section 2.9 on error conditions.

As cards are added, they are stored in an online working copy of the rolodex. In order for the rolodex to be permanently saved in a file, the user must invoke the `File->Save` command. When the user selects `File->Save` from the state of the system shown in Figure 5, the screen appears as shown in Figure 6. The file-save dialog appears in a pop-up window, separate from the main Rolodex Tool window. To

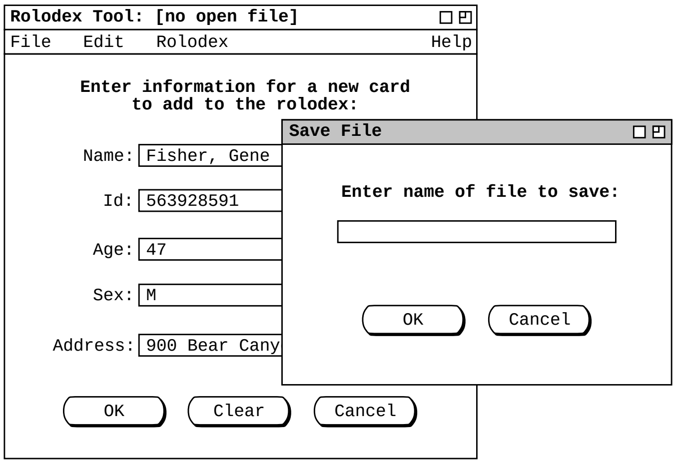

Figure 6: Initial file saving pop-up dialog.

complete the save, the user enters the name of the file and presses `OK`; to cancel the save, the user presses `Cancel`. In either case (`OK` or `Cancel`), the system closes the pop-up save dialog, removing it from the screen. Figure 7 shows the result of the user having confirmed the save onto the file named "`Office Rolodex`". When a rolodex file is saved, the name of the file is displayed in the banner of the tool window, as shown in the figure. Further details of the file commands are covered in Section 2.6.

2.3. Finding Cards

When the user selects the `Find` command from the `Rolodex` menu, the system updates the display area as shown in Figure 8. To locate the desired card(s) the user types in a name and presses the `OK` button. In order for the `Find` to succeed, the name entered in the dialog must be an exact match of a name that appears on one or more cards in the rolodex. For example, if a card is entered with the `Add` command

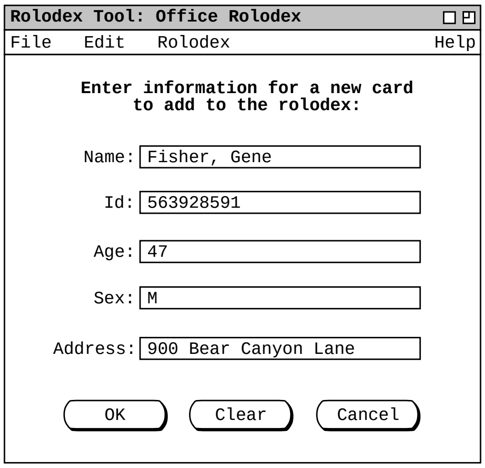

Figure 7: Confirmed initial save.

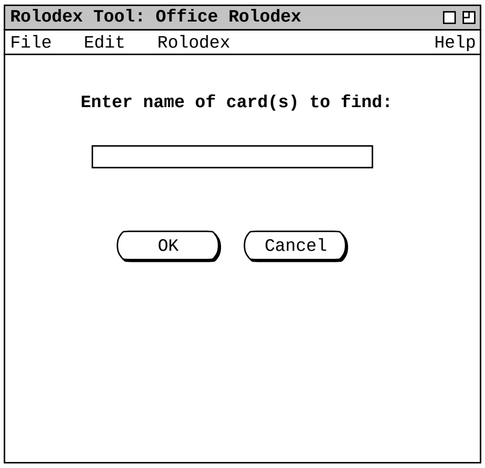

Figure 8: Find card dialog.

with the name "`Fisher, Gene`", then in order for `Find` to locate the card successfully, the user must enter this exact name in the `Find` dialog. Exact match means that two names agree in all punctuation, spacing, and the upper/lower case spelling of all letters. Given these requirements, none of the following names matches "Fisher, Gene": "`Gene Fisher`", "`GENE FISHER`", "Fisher, Gene", "Fisher, G".

To cancel the find command entirely, the user presses `Cancel` in the dialog. When `Cancel` is pressed, the entire data area is cleared, leaving it empty as in the initial rolodex display shown in Figure 1.

There are three possible outcomes when the user presses `OK` in a find–card dialog:

a. zero cards of the given name are found
b. exactly one card of the given name is found
c. two or more cards of the given name are found

Each of these outcomes results in a different output display.

Suppose, for example, that the rolodex contains no cards with the name "Smith, John" and the user inputs this name as shown in Figure 9. In this case, when the user presses `OK` the system finds zero cards and the resulting display contains the simple explanatory message shown in Figure 10. When the user presses the `OK` button in the no-cards-found dialog, the system restores the display to the find–card dialog that resulted in no cards being found, e.g., Figure 9 in this case.

When the system finds exactly one card, the information for that card is shown in the display area. For example, suppose the rolodex contains one card with the name "`Fisher, Gene`", and the user enters this name as shown in Figure 11. In this case when the user presses `OK`, the system displays the information for the found card, as shown in Figure 12. The displayed information is that previously entered using the `Add` command, as shown in Figure 5.

In any found-card display, such as Figure 12, none of the card fields is editable. That is, each of the five value display boxes is read only. Hence the only active element in the display area is the `OK` button. When the user presses this `OK` button, the system restores the display to the dialog that resulted in the card being found, e.g., Figure 11 in this case.

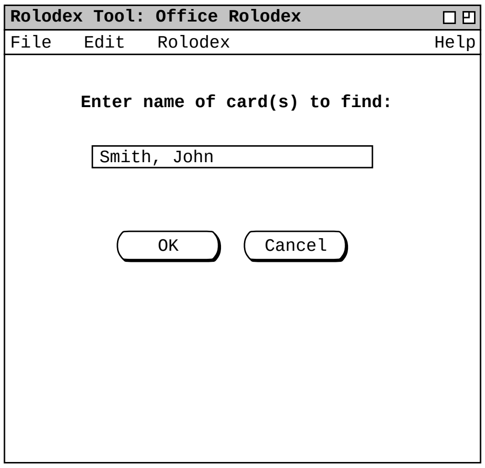

Figure 9: Find card dialog filled in with name appearing on no card in the rolodex.

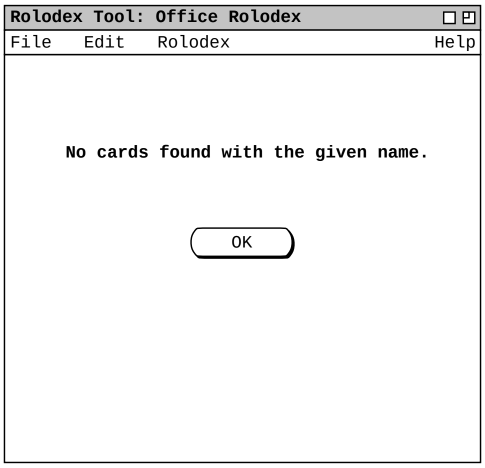

Figure 10: No cards found.

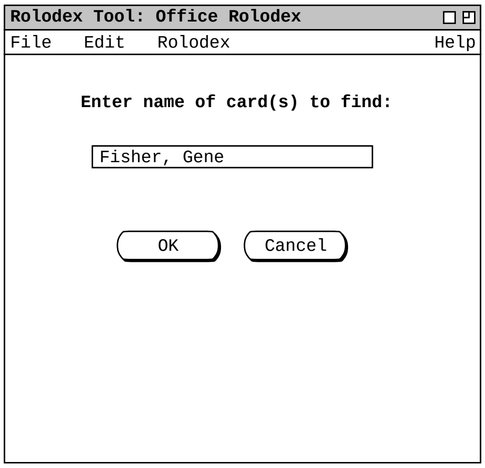

Figure 11: Find card dialog filled in with name appearing on exactly one card.

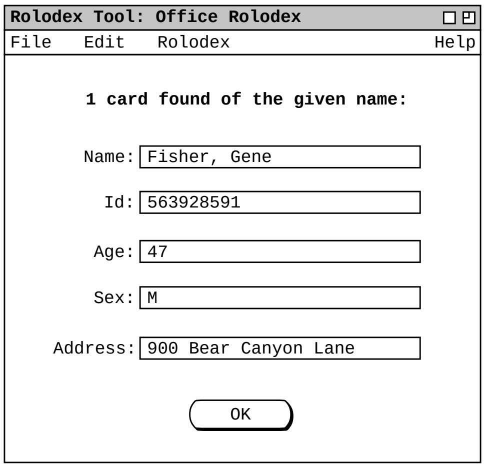

Figure 12: One card found.

When the system finds multiple cards, it displays a selectable list of the found cards. For example, suppose the rolodex contains three cards with the name "Doe, Jane". If the user searches for this name, the resulting display appears as shown in Figure 13. The user can traverse through the list of multiple cards using the `Next` and `Previous` buttons. Each time the user pressses `Next`, the system displays the next found card in the data area. Pressing `Previous` displays the previous card. When the card at the end of the list is reached, pressing the `Next` button has no effect, i.e., the display is unchanged. Similarly, when the users presses `Previous` at the beginning of the list, the display is unchanged. When the user presses the `Done` button, the system restores the display to the find–card dialog that resulted in the cards being found (e.g., the seach dialog for "`Doe, Jane`" in this case).

The order of the cards in the multi-card list is by ascending Id field. Since the Id field is unique for each card, the sorting order is unique. As with the single card display, none of the card data fields is editable in any of the displayed cards.

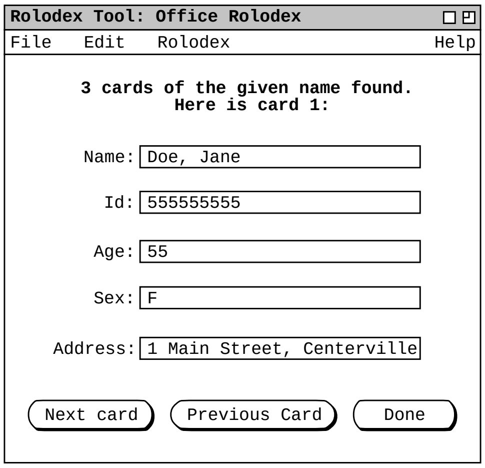

Figure 13: Multiple cards found.

2.4. Changing Cards

To change or delete an existing card, the user must first locate the card by name, as with the `Find` command. Once the card has been located, the `Change` and `Delete` commands behave in a similar fashion.

When the user selects `Change` from the `Rolodex` menu, the system updates the display area as shown in Figure 14. In response to the change–card dialog, the user enters the name of a card to be changed. As with the `Find` command, the name must be entered exactly as it appears on an existing card in the rolodex. For example, to change the card with name, "`Fisher, Gene`", the user types the name as shown in Figure 15.

To cancel the `Change` command entirely, the user presses `Cancel` in the dialog. When `Cancel` is pressed, the entire data area is cleared, leaving it empty as in the initial rolodex display shown in Figure 1.

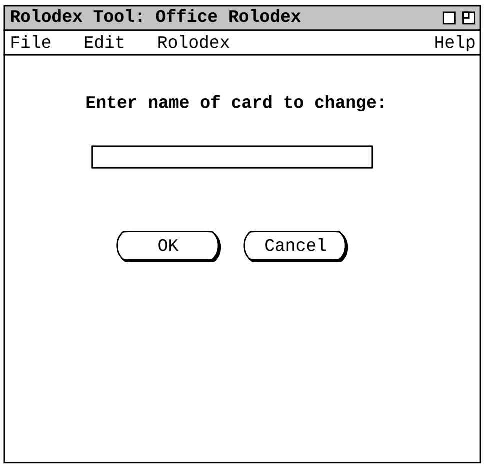

Figure 14: Change card dialog.

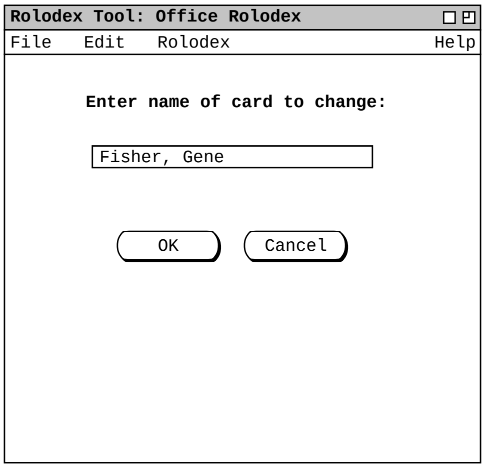

Figure 15: Change card dialog filled in.

As with `Find`, there are three possible outcomes when the users presses `OK` in the change–card dialog: zero, one, or multiple cards found. When the system finds zero cards for the `Change` command, it displays exactly the same dialog as for `Find` (Figure 10). When the user presses `OK` in the no-cards-found dialog, the system restores the display to the change–card dialog that resulted in no cards being found.

When the system finds exactly one card, the display contains the information for that card, as shown in Figure 16. At this point, the interface for `Change` behaves differently than that for `Find`. Specifically, instead of the message "`1 card found of the given name`", the message is "`Change one or more card values:`". To perform the desired changes, the user edits one or more card field values. For example, Figure 17 shows the result of the user having edited the age and address fields of the card found in Figure 16. Any and all fields of a card may be changed, as long as the normal conditions of the `Add` command are met. These conditions are covered in Section 2.8 on data entry details.

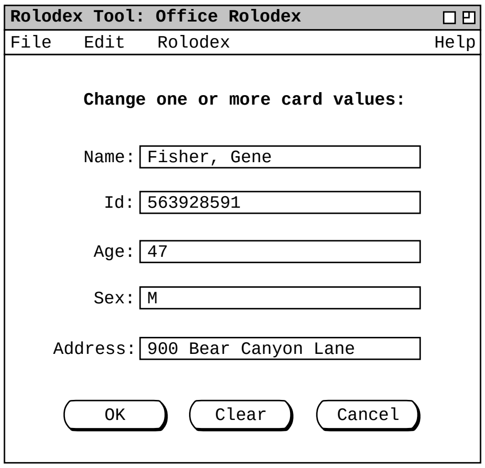

Figure 16: One card found to change.

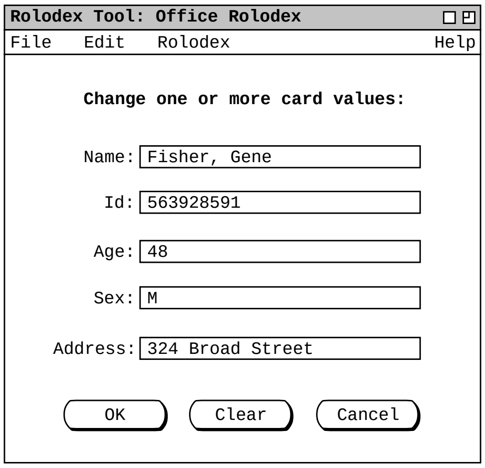

Figure 17: Changes made to a card.

To confirm the changed card into the rolodex, the user presses the OK button at the bottom of the dialog. When the user presses OK, the change occurs and display does not change. That is, the typed-in information stays in place. Although it is not typically likely, the user may make additional changes to the card and press OK as many additional times as desired. In contrast to the Add command, successive presses of OK in the change dialog do not add successive new cards. Rather, only a single card is successively changed. In effect, the Change command removes an existing card, and replaces it with a newly added card, so the total number of cards in the rolodex is the same before and after a change is executed. Pressing OK without making any changes to card field values has no effect on the rolodex.

To clear all of the typed information, the user presses the Clear button. To cancel the Change command entirely, the user presses Cancel. When Cancel is pressed, the display is restored to the change–card dialog that resulted in the original card being found. For example, if the user presses Cancel in the dialog of Figure 16 the system restores the display to the state shown in Figure 15.

The Change command can be applied to only one card at a time. If the system finds multiple cards for a given name, the user must first select a single card to be changed. For example, Figure 18 shows the user having entered the name "Doe, Jane" in the initial change–card dialog. Assuming there are three rolodex cards with this name, when the users presses OK in the dialog of Figure 18, the display appears as shown in Figure 19. The multi-card dialog for Change is similar to that for Find, as shown in Figure 13. The differences are in the message at the top of the dialog and two buttons labeled "Select" and "Cancel" instead of the single Done button. The Next and Previous buttons work as they do in the multi-card Find dialog. The user presses these buttons to locate a specific card to change. When the card is located, the user presses the Select button to proceed with the change of that card. For example, if the user selects the second "Jane Doe" card of three, the screen appears as shown in Figure 20. At this point, the user proceeds in precisely the same manner as when a single card is found to change.

Prior to the press of Select in the multi-card change list, the card fields are read only. The user must press Select in order to proceed with change editing. Alternatively, the user may press Cancel to cancel the Change command entirely. When Cancel is pressed, the display is restored to the change–card dialog that resulted in the original cards being found. For example, if the user presses Cancel in the dialog of Figure 19 the syetem restores the display to the state shown in Figure 18.

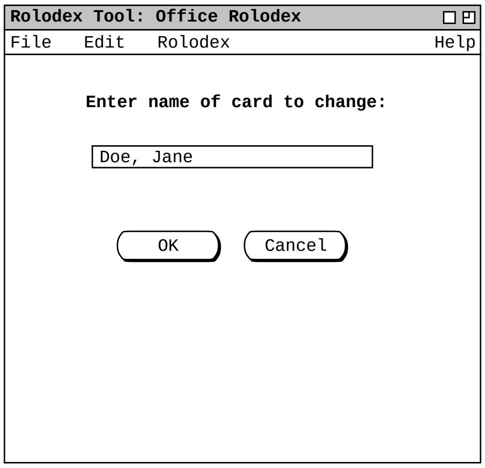

Figure 18: Change card dialog filled in with name to be found on three cards.

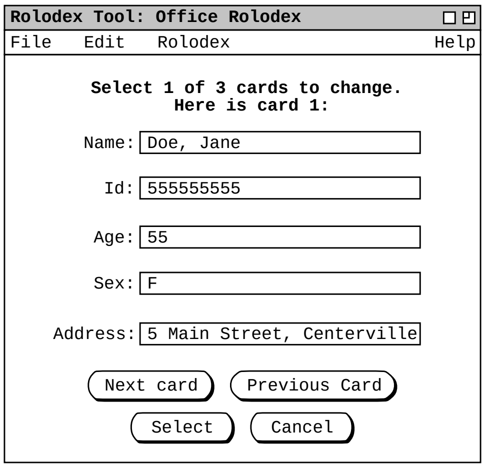

Figure 19: Multiple cards found to change.

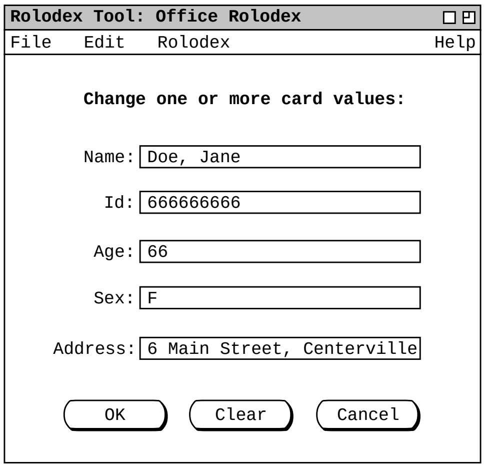

Figure 20: Second of three change cards selected.

2.5. Deleting Cards

The `Delete` command behaves much the same as the `Change` command, with appropriate differences in the dialogs. To clarify the similarities and differences between `Change` and `Delete`, the scenario for `Delete` in this section parallels the scenario for `Change` in the preceding section.

When the user selects `Delete` from the `Rolodex` menu, the system updates the data area as shown in Figure 21. In response to the delete–card dialog, the user enters the name of a card to be deleted. As with both the `Find` and `Change` commands, the name must be entered exactly as it appears on an existing card in the rolodex. For example, to delete the card with name, "`Fisher, Gene`", the user types the name as shown in Figure 22.

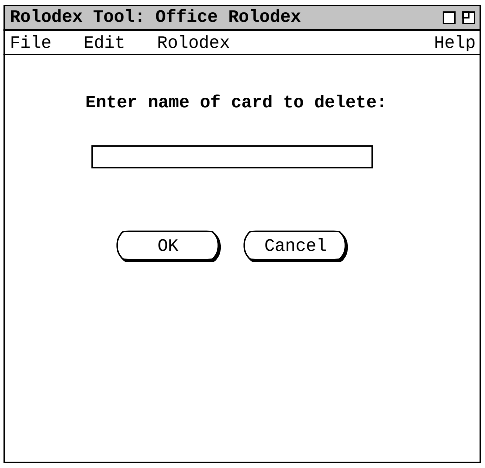

Figure 21: Delete card dialog.

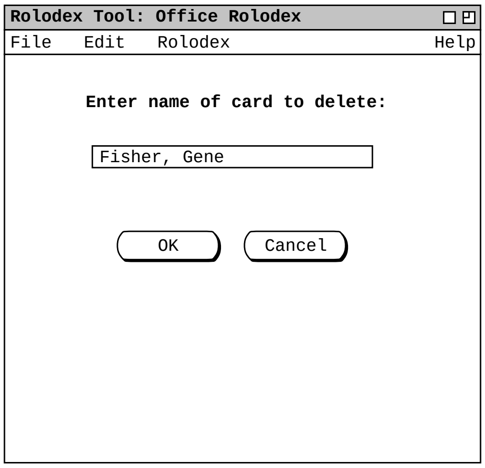

Figure 22: Delete card dialog filled in.

To cancel the `Delete` command entirely, the user presses `Cancel` in the dialog. When `Cancel` is pressed, the entire data area is cleared, leaving it empty as in the initial rolodex display shown in Figure 1.

As with `Find` and `Change`, there are three possible outcomes when the user presses `OK` in the delete–card dialog: zero, one, or multiple cards found. When the system finds zero cards for the `Delete` command, it displays exactly the same dialog as for `Find` (Figure 10). When the user presses `OK` in the no-cards-found dialog, the system restores the display to the delete–card dialog that resulted in no cards being found.

When the system finds exactly one card, the display contains the information for that card, as shown in Figure 23. At this point, the interface for `Delete` behaves differently than that for `Find` or `Change`. Specifically, instead of the `Find` or `Change` message, the message is "`Delete the following`

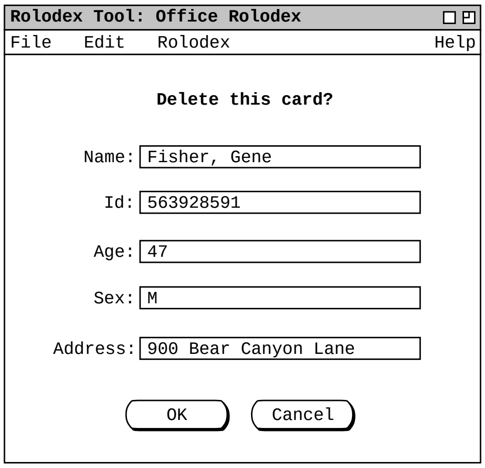

Figure 23: One card found to delete.

card?". To confirm the delete, the user presses the OK button. When OK is pressed, the card is deleted and the entire data area is cleared, leaving it empty as in the initial rolodex display shown in Figure 1.

If the user presses Cancel in the delete confirmation dialog, the display is restored to the original delete–card dialog that resulted in the card being found, e.g., Figure 22 in this case.

As with Change, the Delete command can be applied to only one card at a time. If the system finds multiple cards for a given name, the user must first select a single card to be deleted. For example, Figure 24 shows the user having entered the name "Doe, Jane" in the initial delete–card dialog. Assuming there are three rolodex cards with this name, when the users presses OK in the dialog of Figure 24, the display appears as shown in Figure 25. The multi-card dialog for Delete is essentially the same as that for Change shown in Figure 19. The only difference is the message at the top of the dialog that says "Select 1 of N cards to delete" instead of "Select 1 of N cards to change".

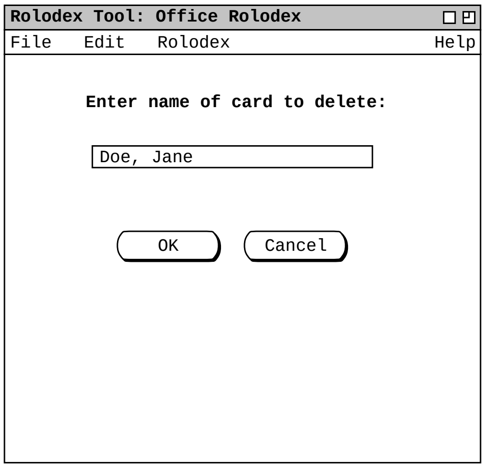

Figure 24: Delete card dialog filled in with name to be found on three cards.

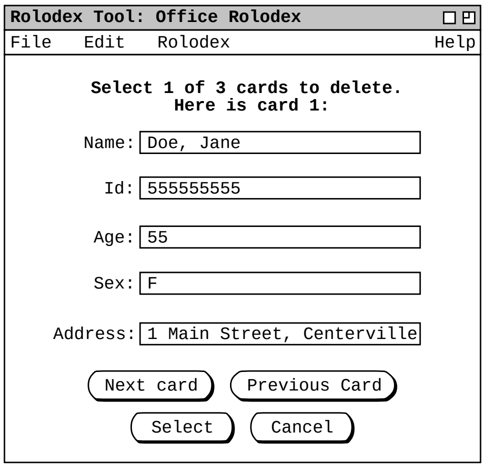

Figure 25: Multiple cards found to delete.

As in the multi-card change dialog, the card fields are read only. The `Next` and `Previous` buttons work exactly as they do in the multi-card `Change` dialog. That is, the user presses these buttons to locate a specific card to delete. When the card is located, the user presses the `Select` button to proceed with the delete. For example, if the user selects the second `"Jane Doe"` card of three, the screen appears as shown in Figure 26. At this point, the user proceeds in precisely the same manner as when a single card is found to delete.

The user may press `Cancel` at any point in the multi-card display to cancel the `Delete` command entirely. When `Cancel` is pressed in the multi-card list, the display is restored to the delete–card dialog that resulted in the cards being found. For example, if the user presses `Cancel` in the dialog of Figure 25 the system restores the display to the state shown in Figure 24.

In any delete–card display, none of the card fields is editable. That is, each of the five value display boxes is read only.

2.6. File and Edit Commands

The commands available on the `File` and `Edit` menus provide an interface between the Rolodex Tool and the underlying operating environment, including its file system. Different operating environments have potentially varying rules for the processing of these commands. The requirements here are generic and operating-environment independent.

2.6.1. Tool Invocation with and without a Rolodex File

The Rolodex Tool can be invoked both with and without a Rolodex file. Without a file, the tool begins execution with an empty Rolodex workspace containing no cards. The banner of the display window indicates there is no open file, as shown in Figure 1. When invoked with an initial file, the tool begins execution with a workspace containing the cards of the specified file. The banner of the display window indicates the open file by name, as shown for example in Figure 7. The tool may not be open with more than one file at one time.

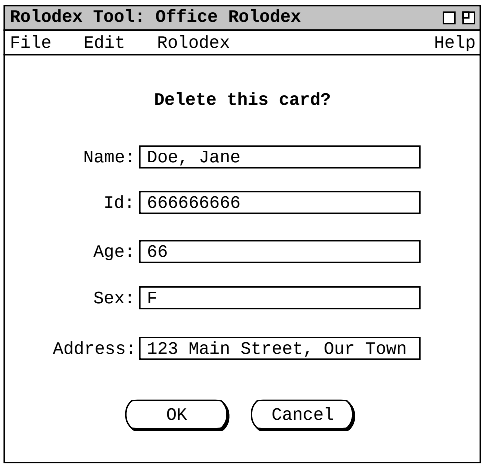

Figure 26: Second of three delete cards selected.

Depending on the operating environment in which the tool is installed, it can be invoked by name from a command line or by some other means such as operating system command menu, Rolodex tool icon, or Rolodex file icon. The means to specify an initial file is also dependent on the operating environment. Two typical means are typing a file name on the command line, or clicking on the icon of a Rolodex file.

2.6.2. New and Open

The `New` command creates a new empty rolodex. When the user selects the `New` command from the `File` menu, the system first checks if the currently active workspace has been saved since the the most recent updating command. If the workspace requires saving, the system displays the dialog shown in Figure 27. The offer-to-save dialog appears in a pop-up window separate from the main Rolodex Tool window. To indicate that the file should be saved, the user presses the `Save` button. At this point, the system proceeds with a `Save` command, as described in Section 2.6.3.

If the `Save` command executes successfully or the workspace did not require saving, the system proceeds to execute the `New` command by clearing the Rolodex workspace, leaving it empty of any cards. The file indicator in the window banner is changed to `"[no open file]"` and the offer-to-save dialog is closed. Figure 28 shows the result, which is exactly the same screen as when the Rolodex Tool is initially invoked without a file.

If the user presses `Discard` in the offer-to-save dialog of Figure 27, the system proceeds with the `New` command by clearing the workspace, changing the file indicator banner, and closing the pop-up dialog. All unsaved changes to the Rolodex are lost.

If the user presses `Cancel` in the offer-to-save dialog, the file is not saved and the `New` command is cancelled, leaving the state of the Rolodex workspace unchanged. The same result obtains if the user cancels the `Save` command that is initiated by pressing the `Save` button in the offer-to-save dialog.

The `Open` command operates in a similar fashion to `New`. When the user selects `Open` from the `File` menu, the system checks the changed status of the workspace and displays the offer-to-save dialog of

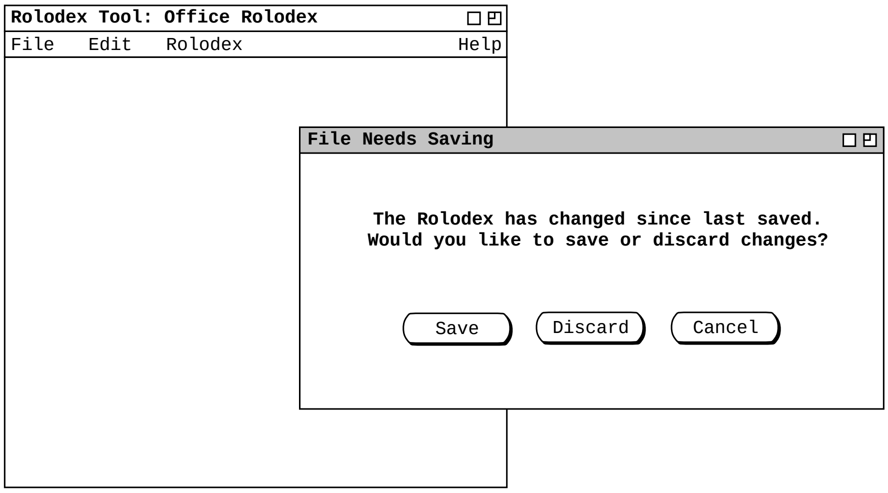

Figure 27: Offer-to-save dialog.

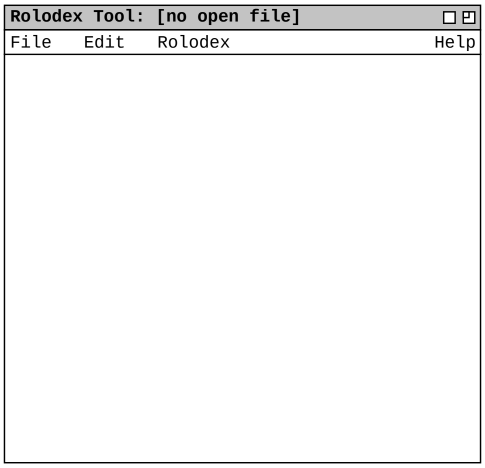

Figure 28: Successful `New` command.

Figure 27 if the workspace requires saving.

If the `Save` command executes successfully or the workspace did not require saving, the system proceeds to execute the `Open` command by displaying the dialog shown in Figure 29 To complete the open, the user enters the name of the desired file in the dialog text box, as show in Figure 30. If the given name is that of an existing and readable Rolodex file, the system responds by opening the file, replacing the current contents of the Rolodex workspace with the file contents, changing the display window banner, and closing the open-file dialog. The result is shown in Figure 31. Section 2.9.4 on file access errors covers the case where the file name to be opened is not a legal Rolodex file or otherwise inaccessible.

The resulting display area for any successful `Open` command is empty, i.e., it contains the display of no card. Hence, the only display difference between a successful `Open` command versus a successful `New`

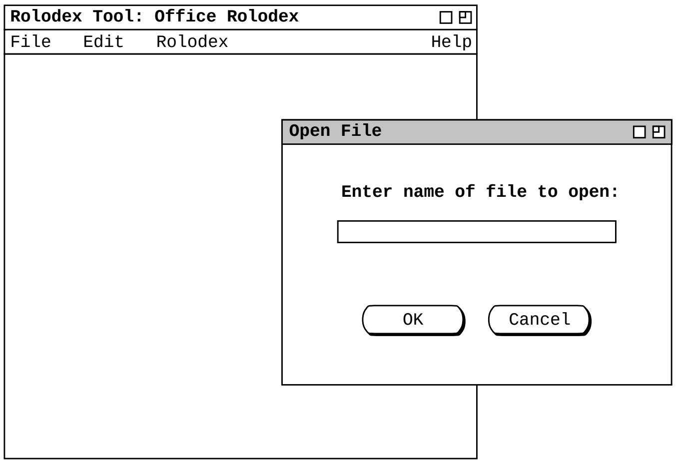

Figure 29: Open dialog.

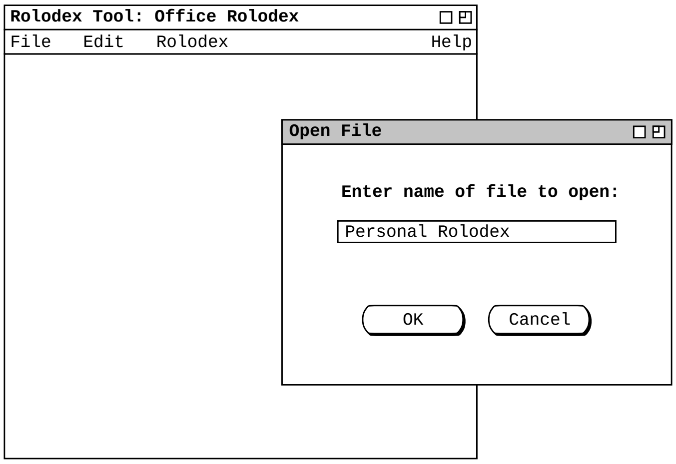

Figure 30: Open dialog filled in.

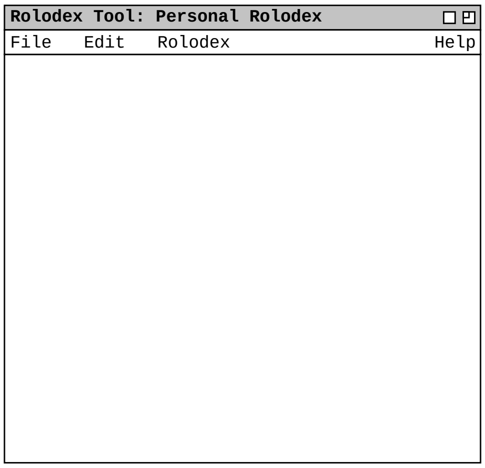

Figure 31: File successfully opened.

command is the file name in the display banner.

If the user presses `Discard` in the offer-to-save dialog, the system proceeds with the `Open` command by displaying the dialog of Figure 29. From this point, the user proceeds as described above, the only difference being that if a file is opened, any unsaved changes to the previously open Rolodex are lost.

If the user presses `Cancel` in the offer-to-save dialog, the file is not saved and the `Open` command is cancelled, leaving the state of the Rolodex workspace unchanged. The same result obtains if the user cancels the `Save` command that is initiated by pressing the `Save` button in the offer-to-save dialog.

For both the `New` and `Open` commands, the precise definition of "requires saving" is based on whether the current state of the Rolodex workspace has changed since the last confirmed execution of a `Save` command. Specifically, the state of the Rolodex is defined as *changed* whenever the user executes a *confirmed updating command*. An *updating command* is defined as one of the three `Rolodex` commands `Add`, `Change`, or `Delete`. The execution of an updating command is *confirmed* when the user presses the OK button in a command confirmation dialog and the command completes execution without error. The confirmation dialog for the Add command is that shown in Figure 5 of Section 2.2. The confirmation dialog for the Change command is that shown in Figures 17 and 20 of Section 2.4. The confirmation dialog for the Delete command is that shown in Figures 23 and 26 of Section 2.5.

This definition of "requires saving" implies the following behaviors for the New and Open commands:

a. If the user executes New two or more times in succession without an intervening confirmed update, the second and subsequent executions of New have no effect.

b. For the purposes of the Open command, execution of New does not mean that the state of the rolodex requires saving, just that the rolodex workspace is empty. If the user executes Open on an empty workspace, the offer-to-save dialog is not displayed.

2.6.3. Save and Save As

When the user selects the Save command from the File menu, the system first checks if there is a currently open file. To the user, the currently open file is indicated by its name in banner of the Rolodex Tool main display window. If there is no current file, the string "[no open file]" appears in the banner.

If there is an open file, the system responds to the Save command by saving the current contents of the rolodex workspace onto the current file. No further user interaction is required in this case, and the state of the display screen does not change.

If there is no current file when the user selects Save, the system displays the dialog shown in Figure 32. To complete the save, the user enters the name of the file and presses OK; to cancel the save, the user presses Cancel. In either case, the system closes the pop-up save dialog, removing it from the screen. Figure 33 shows the result of the user having confirmed the save onto the file named "New Rolodex". The only change to the state of the main display window is the file name in the banner having been changed to "New Rolodex". In particular, the contents of the display area are unchanged. E.g., if the display area was empty prior to a Save, then it is empty afterwards. If the display area contained some command dialog prior to a Save, then the same dialog (in the same state) is in the display area

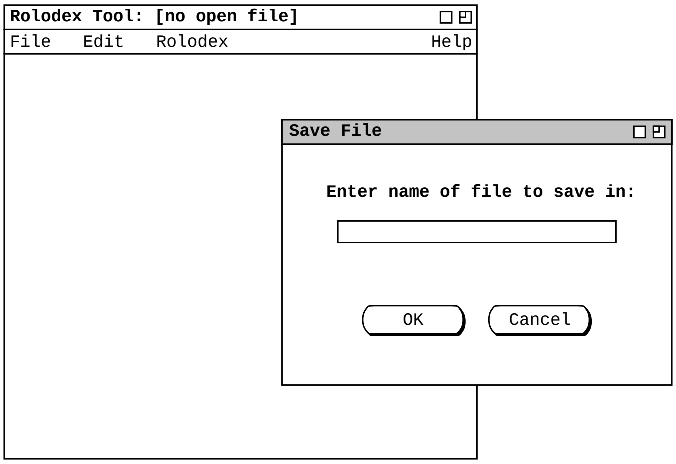

Figure 32: File save dialog, from Save with no open file or from Save As.

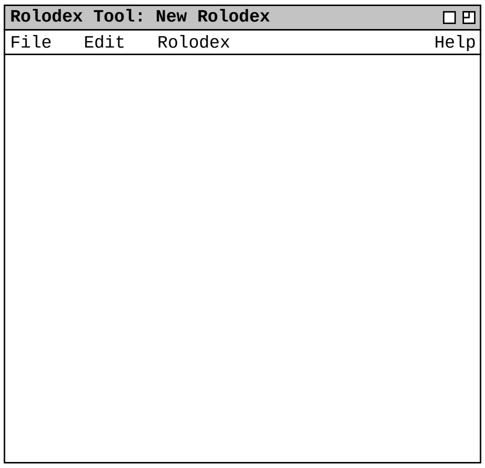

Figure 33: Confirmed save.

afterwards.

When the user selects the `Save As` command from the `File` menu, the system displays the dialog shown in Figure 32 The user interacts with this dialog in exactly the same manner as described above for the `Save` command when there is no open file. The results of a successful `Save As` command are to write the contents of rolodex workspace to the specified file and to change the file name in the display window banner. The data area display is unchanged.

A Rolodex workspace of any size, including zero cards, can be saved. To allow saving an empty workspace, the definition of "requires saving" for `Save` and `Save As` is slightly different from the definition of "requires saving" given earlier for `New` and `Open` (at the end of Section 2.6.2). Specifically, for `Save` and `Save As`, "requires saving" means that the workspace has changed (by the earlier definition) *or* the workspace is empty and has not yet been saved. By this revised definition, `Save` and `Save As` only have an effect if the current workspace requires saving. Otherwise, `Save` and `Save As` have no effect.

This revised definition of "requires saving" implies the following specific behaviors for `Save` and `Save As`:

a. If the user executes `Save` two or more times in succession without an intervening confirmed update, the second and subsequent executions of `Save` have no effect.
b. For the purposes of `Save` and `Save As`, execution of `New` *does* mean that the state of the rolodex requires saving. This is in contrast to what `New` means for `Open` as described earlier. I.e., for the purposes of `Open`, execution of `New` does *not* mean that the workspace requires saving.

Exceptional and error conditions that can arise during execution of the `Save` and `Save As` commands are covered, respectively, in Section 2.8.3 on file accessibility and Section 2.9.4 on file access errors.

Details of file−choosing dialogs vary widely in different operating environments. Typically, a file chooser interface is more sophisticated than the simple name-entry dialog shown in Figure 32. Implementors of the Rolodex Tool must provide the file command functionality described in this section via an interface appropriate to the operating environment. In particular, if the operating environment has standards or conventions for how application programs interface with file commands, then implementors must follow those standards or conventions.

2.6.4. Print

When the user selects the `Print` command from the `File` menu, the system prints a list of cards, sorted alphabetically by name. Cards with the same name are sorted in ascending order by Id field. Figure 34 shows the result of a `Print` command for a rolodex containing five cards. The specific output format rules are as follows:

a. The heading for each of the five card fields ("`Name:`", etc.) is left justified, capitalized, and followed by a colon plus one space character.

b. The value for each of the five card fields is printed exactly as entered by the user in text boxes of the dialogs for adding or changing a card.

c. There is one blank line separating each five-line card in the output list.

d. The sorting of names is based on the lexical comparison of name strings, including punctuation; specifically, the characterwise comparison of name strings is based on the ASCII collating sequence, or comparable collating sequence in standard use in a particular operating environment.

```
Name: Doe, Jane
Id: 555555555
Age: 55
Sex: F
Address: 5 Main Street, Centerville

Name: Doe, Jane
Id: 666666666
Age: 66
Sex: F
Address: 6 Main Street, Centerville

Name: Doe, Jane
Id: 777777777
Age: 77
Sex: F
Address: 7 Main Street, Centerville

Name: Fisher, Gene
Id: 563928591
Age: 47
Sex: M
Address: 900 Bear Canyon Lane

Name: Smith, John
Id: 111111111
Age: 50
Sex: M
Address: 1313 Mockingbird Lane
```

Figure 34: Result of Print command.

Details of file printing vary widely among different operating environments. Such details include printer selection, setup, and other options. Typically, the user interacts with one or more dialogs to specify printing options. Specific details of this interaction are beyond the scope of these requirements. Implementors of the Rolodex Tool must provide the user access to all necessary functionality to accomplish printing successfully in a given operating environment.

The Rolodex Tool provides only the most basic printing facility. To accomplish more sophisticated output formatting, the user may redirect printing output to a file and open the file in a suitable word processor. This assumes that output redirection is an available printing option in the operating environment and that the user has access to a suitable word processor.

2.6.5. Exit

When the user selects `Exit` from the `File` menu, the system first checks if the rolodex workspace requires saving, based on the first definition of "requires saving" given in Section 2.6.2. If the workspace does require saving, the system displays the offer-to-save dialog of Figure 27. If the `Save` command executes successfully or the workspace did not require saving, the system proceeds to execute the `Exit` command by halting the Rolodex Tool program, thereby returning the user to the operating environment from which the Rolodex Tool was invoked.

If the user presses `Discard` in the offer-to-save dialog, the system proceeds with the `Exit` command, with any unsaved changes to the rolodex workspace being lost.

If the user presses `Cancel` in the offer-to-save dialog, the file is not saved and the `Exit` command is cancelled, leaving the Rolodex Tool still running and the state of the Rolodex workspace unchanged. The same result obtains if the user cancels the `Save` command that is initiated by pressing the `Save` button in the offer-to-save dialog.

2.6.6. Undo

When the user selects `Undo` from the `Edit` menu, the system undoes the effect of the most recently confirmed updating command, or the effect of a previously executed `Undo` command. As defined at the end of Section 2.6.2. the *updating commands* are `Add`, `Change`, and `Delete`. For these commands, the precise definition of `Undo` is that the rolodex workspace is restored to the state that existed immediately prior to the confirmed execution of the command. If no confirmed updating command has been executed since the Rolodex Tool was invoked, `Undo` has no effect.

If `Undo` is executed twice in succession, with no intervening confirmed update between the two executions, the effect of the first of the two `Undo` executions is undone. That is, the state of the rolodex is restored to the state that existed immediately prior to the first `Undo`. Third and further successive executions of `Undo`, without an intervening updating command, cause the system to toggle between the state of the workspace with and without the effect of the most recently confirmed update. This behavior means that only a single level of undo is available. That is, only the single most recently executed updating command can be undone.

`Undo` only applies to the three updating commands and itself, not to any other commands. Specifically, `Undo` does *not* apply to:

- `Find`,
- any `File` command,
- any `Edit` command except itself,

- any typing
- any execution of `Cancel` or `Clear` in any interface dialog.

The `Undo` command has no effect at all on the display screen. For example, if `Undo` is executed immediately after an `Add` command, the contents of the add-card dialog remain on the screen, even though the effect of the `Add` command has been undone.

2.6.7. Cut, Copy, and Paste

Operating systems typically provide a "clipboard" area through which cut and paste editing commands operate. This clipboard holds values that are cut, copied, and pasted in tools such as the Rolodex. Hence, the `Rolodex` `Cut`, `Copy`, and `Paste` commands operate through such a clipboard.

To execute the `Cut` or `Copy` command, the user must first select a segment of text in a text box. For example, Figure 35 shows the result of the user having selected all of the text in the `Name` field of an `Add` dialog. Text selection is performed using click and drag with the mouse, or by other environment-specific means such as double or triple clicking in a text box. After text selection, when the user selects the `Cut` command from the `Edit` menu, the system removes the selected text from the edit box and copies the text into the clipboard. Figure 36 shows the result of the user having selected the `Cut` command in the context of Figure 35.

When the user selects the `Copy` command, the selected text is copied to the clipboard, but is not removed from the display in which it is selected and the text remains selected. For example, if the user selects the `Copy` command in the context of Figure 35. the selected text is copied to the clipboard and the resulting display is completely unchanged from its state immediately prior to the execution of `Copy`. I.e., after the copy command, the display appears exactly the same as in Figure 35.

A particular segment of selected text is *unselected* by selecting another text segment or by single-clicking the mouse anywhere in the Rolodex display. Other environment-specific means of unselecting text may be available.

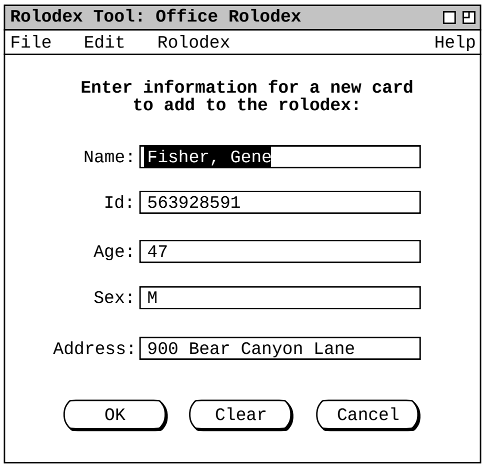

Figure 35: Text selected for Cut or Copy command.

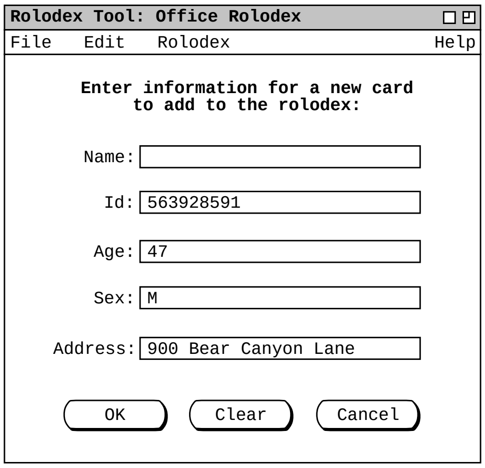

Figure 36: Text cut.

The `Cut` command can only be executed in the context of a display where the text boxes are editable, i.e., where the user may type in the text boxes. In the Rolodex Tool, the text boxes in which the user may *not* type are from the `Find` command, the `Delete` command, and the multi-card display from the `Change` command. These non-editing restrictions are explained, respectively, in Section 2.3, Section 2.5, and Section 2.4

In contrast to `Cut`, the `Copy` command can be executed in any dialog in which text boxes appear, including read-only text boxes. For example, in Figure 37 the user has selected text in a found-card dialog. In this case, the `Copy` command can be executed but the `Cut` command cannot be executed. The `Cut` command as no effect when executed in this context.

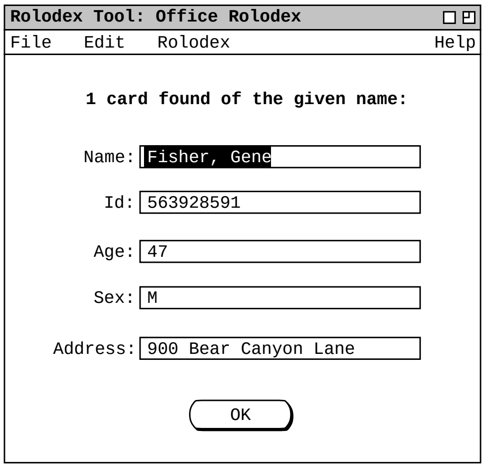

Figure 37: Text selected for Copy only.

To execute the `Paste` command, the user must first execute a `Cut` or `Copy` and then place the typing cursor within some editable text box. After these steps, when the user selects the `Paste` command from the `Edit` menu, the text stored in the clipboard is inserted at the point of the typing cursor, as if it had been entered by the user from the keyboard. Pasted text appears in normal font, i.e., it is not highlighted as selected text.

If the user attempts to execute a `Cut` or `Copy` command without first selecting text, the commands have no effect. If the user attempts to execute a `Paste` command without having executed either a `Cut` or `Copy`, the `Paste` command has no effect. The `Cut`, `Copy`, and `Paste` commands operate only on plain text strings within a single displayed text box. In particular, the commands do *not* apply to a complete card.

The use of the clipboard for communication between the Rolodex Tool and the outside operating environment depends on the capabilities of the environment. Specifically, the environment must support the insertion and removal of plain text to and from the clipboard. If this requirement is met, and the operating environment permits inter-tool clipboard communication, then plain text copied to the clipboard from a another tool can be pasted in the Rolodex using `Paste`. Similarly, plain text cut or copied in the Rolodex Tool can be pasted or otherwise accessed as plain text in another tool, via the clipboard.

2.7. Help Commands

As with the `File` and `Edit` commands, processing of `Help` varies in different operating environments. In particular, some operating environments have specific standards or conventions for the delivery of help information. The requirements here (and in Appendix B) are generic and operating-environment independent. What is specified here is the required content for the help information. This content may be reformatted and possibly reorganized, based on environment-specific conventions.

2.7.1. About

When the user selects `About` from the `Help` menu, the system displays the pop-up dialog shown in Figure 38. The specific contents of the `About` message are subject to change, based on the latest version,

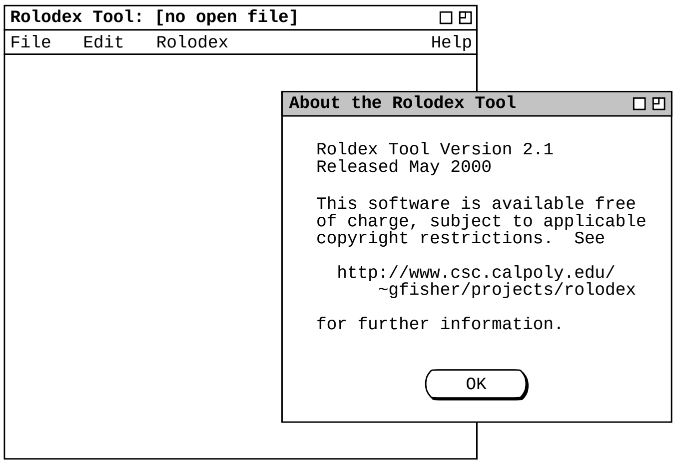

Figure 38: About the rolodex tool.

release date, and repository address. To remove the about message from the screen, the user presses the OK button at the bottom of the About dialog window.

2.7.2. Quick Help

When the user selects `Show Quick Help` from the `Help` menu, the system enters *Quick Help Mode*. In this mode, the system displays a small pop-up message when the user moves the mouse over elements of the display for which quick help is available. For example, Figure 39 shows the result of the user selecting `Show Quick Help` and then moving the mouse over the `File` menu. In quick help mode, the help messages appear with or without any mouse button depressed, whenever the mouse is moved over a display element for which quick help is available. In this way, the user can view quick help for display elements that may appear only when the mouse is depressed, such as menu items. When the user moves the mouse off of an element for which quick help is displayed, the quick help message is removed from the screen.

Figures 40 and 41 show two additional quick help examples, when the user has moved the mouse over the `File->New` menu item and over the `OK` button in the add-card dialog. Whenever quick help is displayed, the help message box appears immediately below the element to which it applies, with the left side of the message box aligned with the horizontal center of the element. Each element for which help is available has a *quick help activation area* defined. This is the area within which the mouse is moved to activate quick help for the element. Geometrically, the quick help activation area is the smallest enclosing rectangle around the screen element for which quick help is available.

After entering quick help mode, the system changes the `Show Quick Help` menu item to `Hide Quick Help`. When the user subsequently reselects the `Help` menu, it appears as shown in Figure 42. When the user selects `Hide Quick Help`, the system exits quick help mode so that quick help messages no longer appear. Each successive selection of the quick-help menu item toggles the menu item text between `Show Quick Help` and `Hide Quick Help`.

The complete list of display elements for which quick help is available is given in Appendix B.1

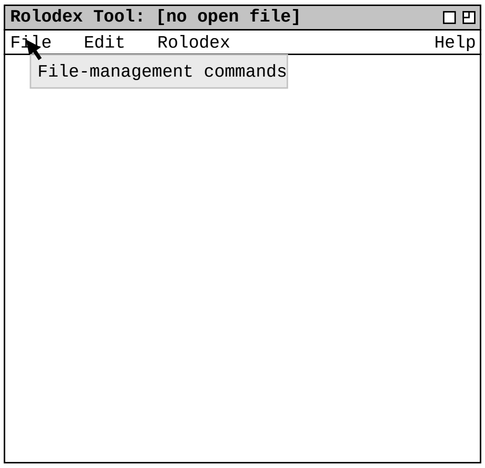

Figure 39: Quick help for the File menu.

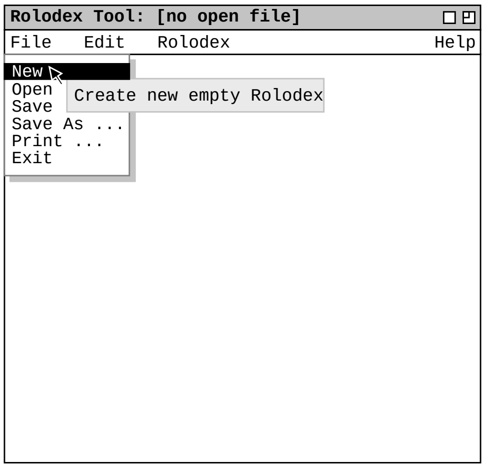

Figure 40: Quick help for File New.

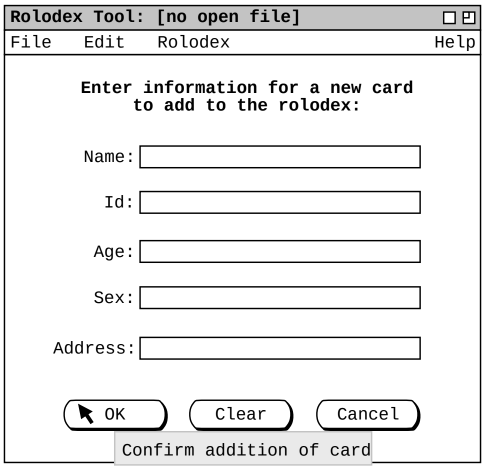

Figure 41: Quick help for add card OK.

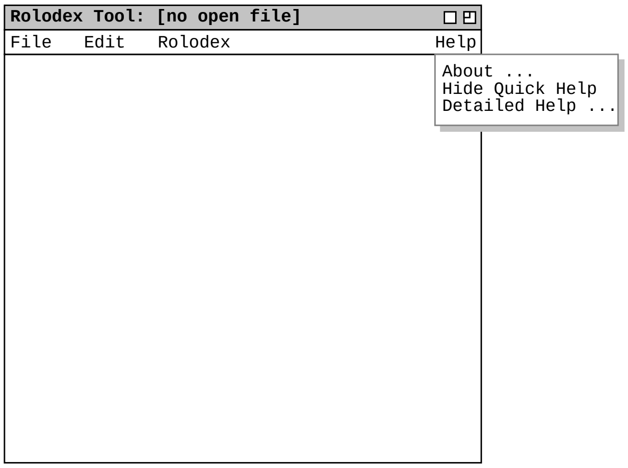

Figure 42: Hide quick help.

2.7.3. Detailed Help

When the user selects `Detailed Help` from the `Help` menu, the screen appears as shown in Figure 43. Detailed help is displayed in a pop-up window, separate from the main Rolodex Tool window. The detailed help window is divided into two horizontal panes. The narrower pane on the left has controls for selecting sections of help information. The wider pane on the right displays the selected information.

The detailed help control panel consists of three tabs for selecting information. The `Contents` tab displays a table of contents for the major topics of information. The `Index` tab displays an alphabetic subject listing. The `Search` tab allows the user to search for a keyword or string that appears anywhere in the help information.

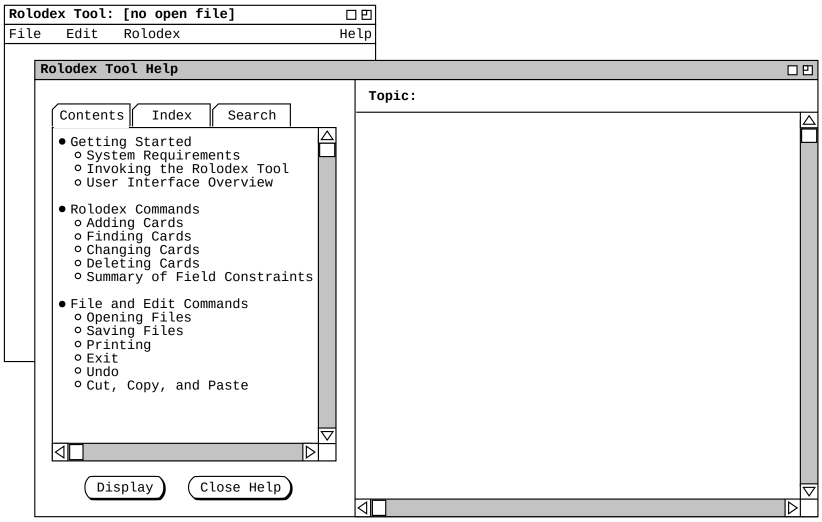

Figure 43: Initial detailed help screen.

The content of detailed help is a reformatted version of the same content that appears in the Rolodex Tool Users Manual in Appendix A. Details of the manual-to-help reformatting are given in Appendix B.2.

2.7.3.1. Detailed Help Contents

The default tab for the detailed help control panel is `Contents`, as shown in Figure 43. The `Contents` pane shows a table of contents for the help information. Each line in the table is a separate topic of information that can be displayed. To select a particular topic for viewing, the user clicks on a single line in the `Contents` pane and presses the `Display` button. For example, Figure 44 shows the result of the user selecting the topic "`User Interface Overview`" in the `Contents` pane, and then pressing `Display`. The selected topic in the `Contents` pane is highlighted and the associated information for that topic is displayed in the information pane on the right. To view another topic, the user selects it in the `Contents` pane and presses `Display` again. In response, the system unhighlights the previously selected topic, highlights the newly selected topic, and displays the information for the newly selected topic. Only a single topic can be selected and displayed at one time. Hence, the extent of the information displayed in the information pane is exactly one topic (if the pane is not empty).

2.7.3.2. Detailed Help Index

When the user selects the `Index` tab in the help control panel, the system updates the help window as shown in Figure 45. The `Index` pane shows an alphabetic list of index terms that appear in the body of the help information. Each line of the `Index` pane is a term consisting of a keyword or key phrase that is described in the help information. To select a term for viewing, the user clicks on a single line in the `Index` pane and presses the `Display` button. For example, Figure 46 shows the result of the user selecting the term "`card field constraints`" and then pressing `Display`. In response, the system displays the information for the topic in which the index term appears, with the textual context of the term highlighted. If necessary, the display is scrolled to the text location where the term appears. The scrolling is performed so that the first line of the paragraph in which the term appears is at the top of the

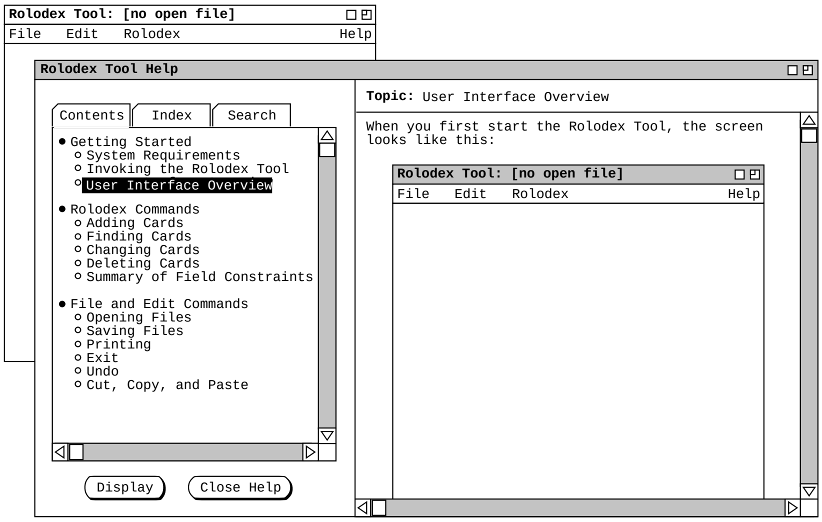

Figure 44: Detailed help for a selected topic.

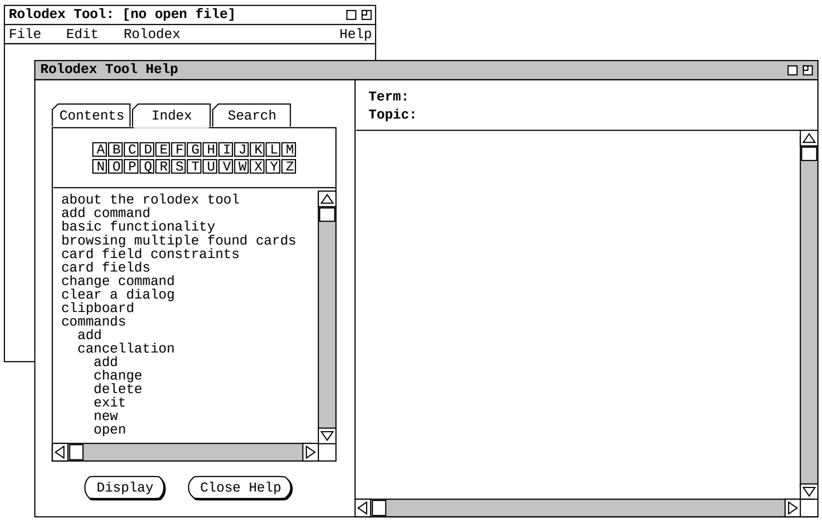

Figure 45: Detailed help index.

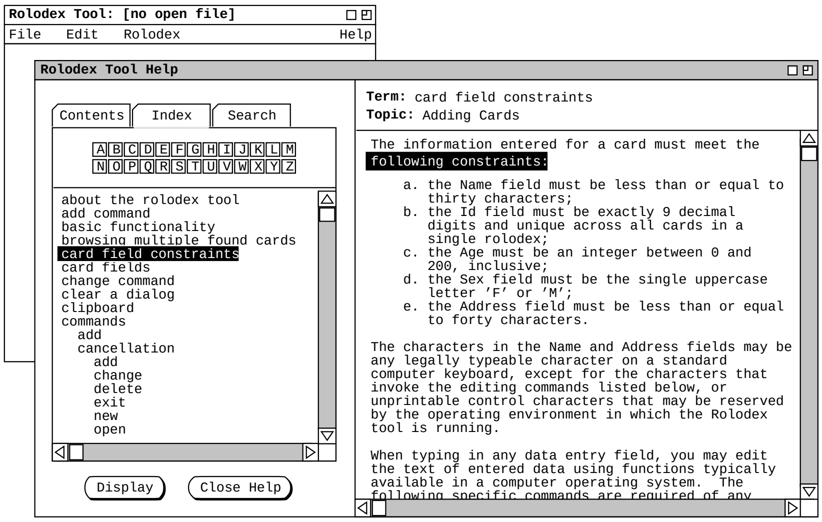

Figure 46: Detailed help for a selected term.

display window.

The information pane for a displayed term contains the information for exactly one topic, where the available topics are those listed under the `Contents` tab. Even if an index term appears more than once in a topic (or topics), the term is linked to a single textual location. Hence, displaying an index term always leads to the same single location in the same topic.

In the `Index` pane of the control panel, the user can scroll through the terms in the normal way using the scroll bar on the right of the pane. The user can also scroll through the index terms by pressing one of the twenty-six small buttons labeled with the letters of the alphabet. When the user presses one of these buttons, the system scrolls the term list so that the first term starting with the selected letter appears at the top of the pane. If there is no index term starting with the selected letter, the system scrolls the list so that the term at the top of the display is the last term starting with the earliest preceding letter for which there is at least one term defined. If there is no such term, the display is unchanged.

2.7.3.3. Detailed Help Search

When the user selects the `Search` tab in the help control panel, the system updates the help window as shown in Figure 47.

The user can search for a string that appears anywhere in the the detailed help information. To perform the search, the user types the desired string in the text box below the "`Enter search string:`" label. For example, Figure 48 shows the result of the user entering the search string "`save`" and then pressing `Display`. After a search, the system displays a list of zero or more topics in which the search string is found. In this example, the string "`save`" is found under the three topics "`User Interface Overview`", "`Adding Cards`" and "`Saving Files`". When a match is found in at least one topic, the system highlights the first topic in the list. In the information pane, the system displays the contents of the highlighted topic, with the first occurrence of the search string highlighted. As necessary, the information pane is scrolled to the text location where the search string appears. The scrolling is performed so that the first line of the paragraph in which the string appears is at the top of the display window.

The extent of a search is the text of all help topics, where the available topics are those listed under the `Contents` tab. The order of the search is from beginning to end through each topic, with topics searched in top-to-bottom order of the `Contents` tab list. If a string occurrence is found in two or more topics, the list of topics is ordered in the relative order of the `Contents` tab. For example, the order of the three topics shown in Figure 48 is the relative order of those topics in the `Contents` tab list.

As with the display of index terms, the information pane for a successful search contains the information for one topic, where the available topics are those listed under the `Contents` tab. In contrast to an index term, a search string may occur in more than one location and in more than one topic. If the string does

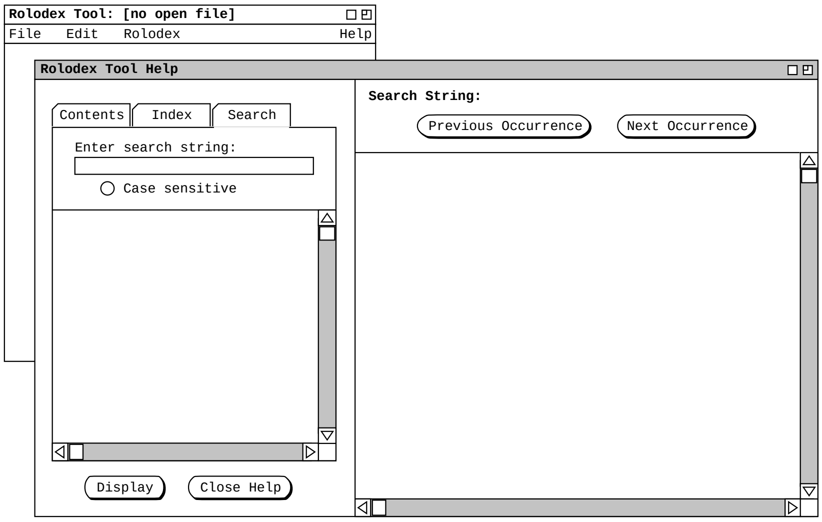

Figure 47: Detailed help search.

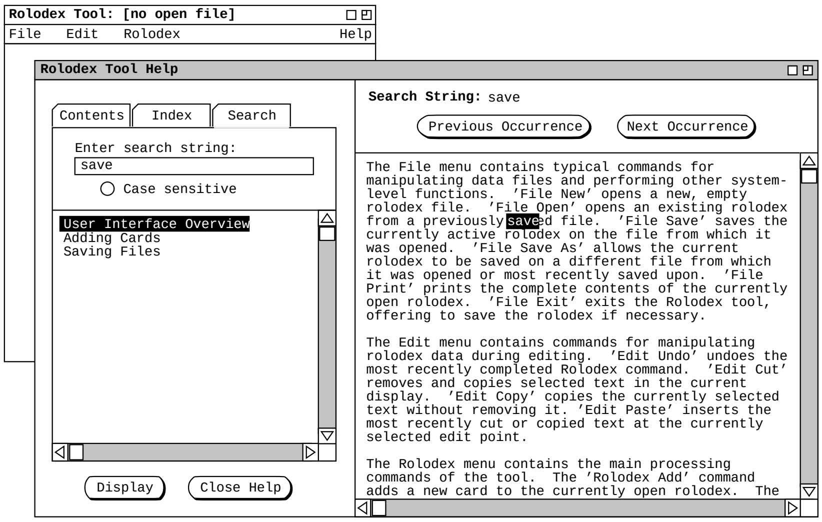

Figure 48: Detailed help for a selected search string.

appear in more than one location, the user employs the `Previous Occurrence` and `Next Occurrence` buttons to view each successive occurrence. As each occurrence is displayed, the string is highlighted and the screen is scrolled so that the first line of the paragraph in which the string appears is at the top of the information display window.

The effect of the previous and next buttons moves across topic boundaries. When the user moves to the last occurrence of a search string in one topic, pressing `Next Occurrence` moves to the first occurrence of the string in the next topic, if any. Similarly, when located at the first occurrence in a topic, pressing `Previous Occurrence` moves to the last occurrence in the preceding topic, if any. Pressing `Previous Occurrence` has no effect when the current occurrence is the first in the first topic. Similarly, pressing `Next Occurrence` has no effect at the last occurrence in the last topic.

As the cross-topic moves take place, the system changes the highlighting appropriately under the `Search` tab. That is, the topic highlighted under the `Search` tab pane is always the topic in which the currently highlighted search occurrence appears in the information pane.

The search string may consist of any typeable characters, including spaces and punctuation. If the radio button labeled "`Case sensitive`" is on, then the search is performed on an exact match basis. That is, an occurrence is considered a match if it is textually identical to the search string, based on all characters in the string. If the `Case sensitive` button is off, then the search is performed on an exact match basis, except alphabetic characters are matched independent of case. Off is the initial value for the the `Case sensitive` button.

2.7.3.4. Help Presentation Details

As the user selects different tabs in the control pane, the information pane always changes accordingly. Specifically:

a. when the `Contents` tab is selected, the information pane contains the "**`Topic`**" heading and displays information for the most recently selected topic from the `contents` pane;

b. when the `Index` tab is selected, the information pane contains the "**`Term`**" heading and displays information for the most recently selected term from the `Index` pane;

c. when the `Search` tab is selected, the information pane contains the "**`Search String`**" heading and displays information for the most recently entered search string.

In all cases, the very first time the user selects a tab, the corresponding information pane is empty except for the heading title.

When the user switches between tabs in the help control panel, the state of the control pane is the same as it was when the tab was most recently selected. Specifically, the most recently selected item (if any) is still highlighted, and the scrolled position of the display is as it was when the tab was most recently selected. For example, if the user selects the `Search` tab in the context of Figure 46 and then reselects the `Index` tab, the resulting display again looks exactly the same as Figure 46.

2.8. Data Entry Details

This section of the requirements defines data entry details noted but not fully defined in the scenarios.

2.8.1. Card Field Values

The information entered for a card must meet the following constraints:

a. the `Name` field must be less than or equal to 30 characters;
b. the `Id` field must be exactly 9 decimal digits and unique across all cards in a single rolodex;
c. the `Age` must be an integer between 0 and 200, inclusive;
d. the `Sex` field must be the single uppercase letter 'F' or 'M';
e. the `Address` field must be less than or equal to 40 characters.

The characters in the Name and Address fields may be any legally typeable character on a standard computer keyboard, except for the characters that invoke the editing commands listed below or unprintable control characters that may be reserved by the operating environment in which the Rolodex Tool is running.

When typing in any data entry field, the user may edit the text of entered data using functions typically available in a computer operating system. The following specific commands are required of any implementation:

a. backward delete of characters with the `delete` or `backspace` key;
b. backward and forward character motion using right and left arrow keys;
c. cursor positioning to the next edit field using the `tab` key;
d. text selection using the mouse.

Additional keyboard functionality, such as using the `return` key for a shortcut to the `OK` button, is not specifically required, but allowed per the conventions of the operating environment in which the Rolodex Tool is running.

2.8.2. Search Strings

Search strings are used in the `Find` command and the `Search` feature of detailed help. In the case of `Find`, searching is performed on a strictly exact match basis. The precise rules for exact string match are the following:

a. the strings have the same number of characters;

b. the characters in the ith position of each string are equal, per the definition of equality defined by the character coding scheme employed in the operating environment in which the Rolodex Tool is running (e.g., the ASCII coding scheme).

In the case of `Search` for detailed help, searching is performed on the basis of case-insensitive exact match or strictly exact match, depending on the setting of the `Case sensitive` selection in the search dialog. The precise rules for case-insensitive exact match are the following:

a. the strings have the same number of characters;

b. the characters in the ith position of each string are equal, per the definition of equality defined by the character coding scheme, *except that* the upper and lower case version of the same letter character are considered equal.

The `Find` and `Search` commands differ with respect to substring matching. In the case of `Find`, substring matching is *not* performed. For *Find*, a full-length card name is both the search string and the target of the search. In order for a match to occur, the length of the search string entered in the find-card dialog must be exactly the same as the length of the name in a rolodex card. Hence, the search name "Smith" does not match the name "Smith, J" in some card, because the search string "Smith" matches only a substring of the target string "Smith, J".

In the case of help searching, substring matching does occur since a search string is compared to multiple targets within the help text. Specifically, the search string is compared against all text substrings of the same length, from the first to the last character positions in the text of a help topic. Hence, for example, the help search string "save" does match in the text "... is saved on ...".

2.8.3. File Access

In order for a file to be used in an `Open` command, it must meet the following constraints:

a. it must exist;

b. it must be readable by the Rolodex Tool user;

c. it must be a legal Rolodex file, produced using the `Save` or `Save As Rolodex` command.

In order for a file to be used in a `Save` or `Save As` command, it must meet the following constraints:

a. it must have a legal file name, per the requirements of the operating environment in which the Rolodex Tool is running;

b. it must be writeable by the Rolodex Tool user.

If the user enters the name of an existing file in the file-save dialog, the system responds with a dialog of the form shown in figure 49. In this figure, the user has entered the name "Office Rolodex", which is assumed already to exist. If the user presses `Yes`, the system proceeds to save the rolodex, overwriting the existing contents of the file. The system then removes both the file-exists and save-file dialogs from the screen. If the user presses `No`, the save command is canceled, the contents of the rolodex workspace are not saved, and the specified file is not changed. In the `No` case, only the file-exists dialog is removed from the screen. The user may then enter a different file name in the save-file dialog.

The Rolodex `File` commands are typical of those available in many user–oriented software tools. Details of the user interface dialogs for these commands can vary widely in different operating environments. The file interface dialogs shown in Section 2.6 are simple and generic, indicating the minimal user interface functionality that must be available in any operating environment. This minimal functionality is the following:

a. Rolodex files are identified by unique name;

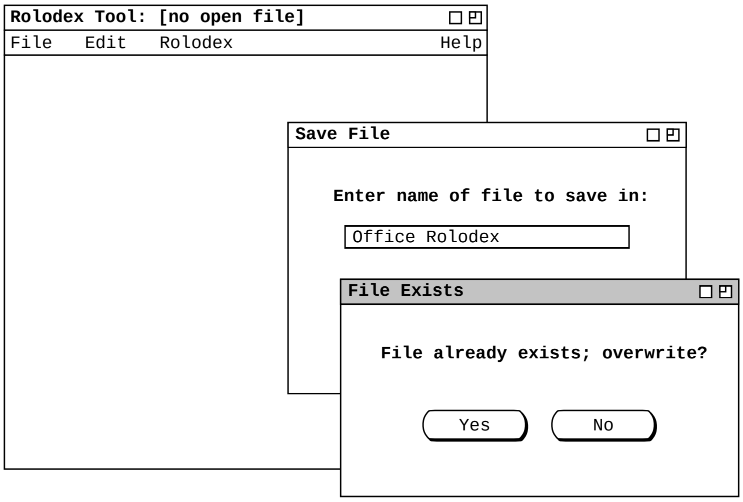

Figure 49: Offer-to-overwrite dialog.

b. the type of a Rolodex file can be distinguished from other file types, such that it can be identified by the Rolodex tool as a valid Rolodex file;

c. files are stored in some form of navigable file system, details of which are implementation-specific.

When implemented in a specific operating environment, the file dialogs may contain additional features, including shortcuts for typing long file names and the means to navigate conveniently in the file space.

2.9. Error Conditions

When the user enters data that do not meet the constraints described in the preceding sections, the system displays an error message dialog of the general form shown in Figure 50. As shown in the Figure, the

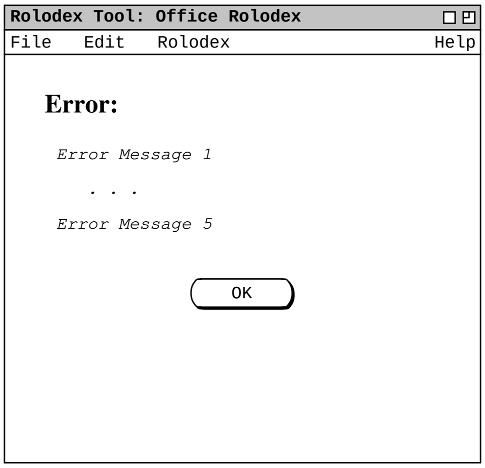

Figure 50: Generic error message dialog.

dialog contains from one to five messages, depending on the specific error(s) being reported. In all cases, there is a single `OK` button at the bottom of the dialog. When the user presses `OK`, the display is restored to its immediately preceding state, that is to the state immediately prior to the error message output. This behavior is uniform for all error messages.

2.9.1. Attempting to Add a Duplicate Card

When the user attempts to add a card with a duplicate `Id`, the system displays the following message in the error dialog:

```
A card of the given Id is in the rolodex.
```

This error condition can arise when the user is executing either the `Add` or `Change` command. As noted in Section 2.2, there are no other constraints on field value duplication. Two or more cards may have duplicate values for any and all fields except `Id`.

2.9.2. Multiple Data Input Errors

Most error conditions are the result of the user typing an illegal value in one or more card fields. When there are multiple errors, i.e., errors in more than one field value, an error message is displayed for each erroneous field. The order of multiple error messages is the same as the order of fields in the card, i.e., `Name` field first through `Address` field last. As an example of the display for multiple error messages, Figure 51 shows the case where the user has entered erroneous values for all five card fields. The Figure shows the specific error message text for each of the five card fields.

The card `Id` is the only field for which two error conditions are possible -- non-uniqueness and incorrect number of digits. These conditions are disjoint, i.e., both conditions cannot apply to the same card. Therefore, a maximum of five error messages can appear for any card.

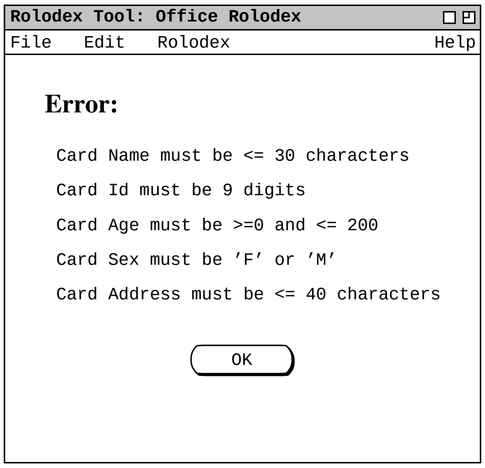

Figure 51: Error input in all five card fields.

2.9.3. Typing in a Read-Only Field

Card fields in certain displays are read only, as described in Sections 2.3, 2.4, and 2.5. When the user attempts to type in a read-only field, the typing does not occur and the system produces a short audible or visible alert (e.g., "bell" sound) for each typed keystroke. The specific sound or visual cue produced for a short alert depends on the operating environment in which the Rolodex Tool is running. If the operating environment allows the user to disable the sound for an audible alert, then the sound produced by the Rolodex Tool is disabled.

2.9.4. File Access Errors Section 2.8.3 defines file access constraints for the `File->Open` and `File->Save` commands. If the user enters a file name that violates the constraints for `Open`, the system responds with a dialog of the form shown in Figure 52. The message informs the user that the selected file cannot be opened. When the user presses the `OK` button, the system removes the cannot-open dialog, whereupon the user can enter a different file name in the open-file dialog, or cancel it.

If the user enters a file name that violates the constraints for `Save` or `Save As`, the system responds with a dialog of the form shown in Figure 53. The message informs the user that the selected file cannot be saved in. When the user presses the `OK` button, the system removes the cannot-save dialog, whereupon the user can enter a different file name in the save-file dialog, or cancel it.

2.10. Graphical User Interface Details

This section of the requirements covers details of the Rolodex Tool graphical user interface (GUI) not fully addressed in preceding sections. Some aspects of the GUI may differ in different operating environments. Such differences are addressed below.

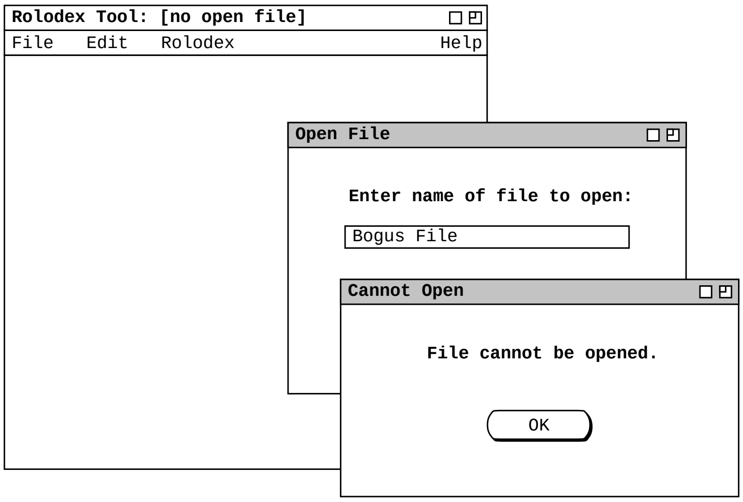

Figure 52: Cannot open file.

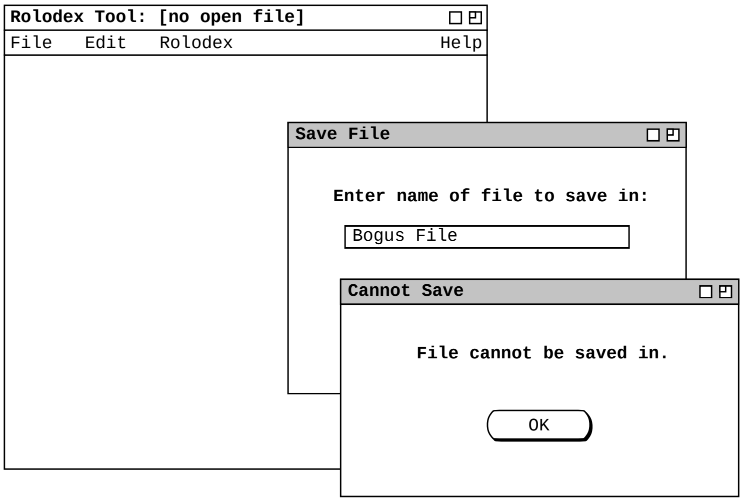

Figure 53: Cannot save file.

2.10.1. Persistent Display States

When the user selects a command from the `Rolodex` menu, the system updates the display to the appropriate dialog for the command. When the system displays a dialog, the state of the dialog is the same as when it was most recently previously displayed. Specifically, the contents of the dialog text boxes are those most recently typed by the user. For example, suppose the user executes the following actions:

a. choose `Rolodex->Add`

b. enter some values in the card fields, with or without pressing `OK`

c. choose `Rolodex->Find`

d. enter a name to find and press `OK`

e. choose `Rolodex->Add` again

At this point, the display shows the add-card dialog with the contents of the text boxes as entered in step b. If the user then proceeds to choose `Rolodex->Find` again, the display shows the find-card dialog with the text box containing the name entered in step d.

2.10.2. Disabling and Enabling Interface Elements

In all of the preceding scenarios, active interface elements are drawn with solid black outlines, without distinguishing between an enabled versus disabled state. In many operating environments, an interface element can be shown in a disabled state, typically by drawing it in a shade of grey. An interface element is shown as disabled when the current state of the system renders the element unusable. For example, the `File->Save` menu item can be shown as disabled when there is no currently open file or when the current file does not require saving.

The requirements given in this document do not explicitly define when an interface element is enabled or disabled. Rather, the requirements define when a user-invoked command has no effect. These "no effect" cases are equivalent to defining an interface element to be disabled. That is, when conditions exist such that a command has no effect, then the interface element that is used to invoke that command can be defined as disabled. Based on this definition, implementors may choose to display interface elements in a disabled state when the system state is such that activating the elements has no effect. The display of disabled interface elements must be consistent for all interface elements. That is, visual enabled/disabled cuing must be used consistently for all interface elements, or for no interface elements at all.

2.10.3. Fonts and Colors

In all of the preceding scenarios, a 12-point Courier font is used for text and all interface elements are drawn in black and white. In some operating environments, the user is able to change the font properties and coloring used in an application program. Where this is possible, such changes can be made in the Rolodex Tool. The following font and color properties are required of any implementation:

a. the relative size of text must be retained; i.e., the font size may not differ across display windows;
b. the distinction between bold face and normal face font must be as shown in the interface scenarios;
c. the default pen color for text and lines must be black, or a dark-shaded color consistent with the conventions of the operating environment in which the Rolodex Tool is running;
d. the default background color for all screens must be white, or a light-shaded color consistent with the conventions of the operating environment.

2.10.4. General Look-and-Feel Issues

In all of the preceding scenarios, interface elements are shown in a plain and simple style. In many operating environments, interface elements are drawn with graphical shading and other forms of graphical embellishment to render them more visually useful and appealing to the user. Implementors may follow the look-and-feel conventions of a particular operating environment for the Rolodex Tool interface. However, the fundamental operational character of the specified interface elements must be implemented as shown in the requirements scenarios. Specifically:

a. the top-level commands must be presented in a menubar and menus;
b. all command buttons must be labeled and justified as shown;
c. all text labels must be spelled, capitalized, and justified as shown;
d. horizontal and vertical text scrollbars must be available as shown (but scrollbars may be removed when not needed).

2.10.5. Help Interface

Some operating environments provide a standard means for users to access quick help or a form of help that is essentially the same as quick help. Viz., it is a very brief text string associated with a particular region of the screen, most typically a region of the screen with which the user interacts. Implementors may follow the conventions of a particular operating environment for the precise format and position of the GUI elements in which the quick help text is displayed.

The following are required aspects of quick help in any implementation:

a. the quick help text defined in Appendix B.1. is delivered precisely as that text;
b. the help is delivered in precisely those mouse contexts defined in Appendix B.1;
c. quick help is explicitly enabled and disabled by the user.

The last of these requirements means in particular that quick help is *not always on*, as is the case in some popular software applications.

Detailed help delivery also differs among operating environments. Implementors may again follow environment-specific conventions for the delivery of detailed help. The following are required aspects of detailed help in any implementation:

a. the detailed help text defined in Appendix B.2 is delivered precisely as that text;

b. the fundamental organization of detailed help must be followed insofar as possible in any operating environment.

When the detailed help conventions of an operating environment differ substantially from those given in Section 2.7 and Appendix B, implementors must consult with the requirements team to ensure that the intent of the requirements for the help commands are met.

3. Non-Functional Requirements

The non-functional requirements for the Rolodex Tool are organized into three major categories:

- system-related,
- process-related, and
- personnel-related

In general, the non-functional requirements for the Rolodex Tool are limited, given the simplicity of tool's functionality and the fact that it is public domain non-commercial software.

3.1. System

System-related non-functional requirements cover performance, operational environment, and general system characteristics.

3.1.1. Performance

Given the limited functionality of the Rolodex Tool and its limited user community, performance is not a significant issue. In general, the speed of all operations must be comparable to similar tools, such as those cited in Section 1.5 on related systems. "Comparable to" means that the Rolodex Tool operations must execute at a speed that is in the same order of magnitude as similar tools under similar computer load conditions.

There are no specific requirements for the allowable size of a rolodex in terms of the number of cards. A Rolodex on the order of thousands of cards is the likely maximum size during normal use. However, there is no specific reason that the size of the Rolodex be limited.

3.1.2. Operational Environment

There are no specific requirements for the hardware or software platform on which the Rolodex Tool must operate. As discussed in Section 2.10 , specific elements of the graphical user interface may differ per the conventions of a specific operating environment.

There are no specific requirements for interoperability with external software, such as other database programs.

3.1.3. General Characteristics

The functionality of the Rolodex tool is sufficiently straightforward that the tool must be 100% reliable, accurate, and correct. "Reliable" means that the system must not fail due to a software fault that is traceable to one of its functional software components. "Accurate" means that the system must not loose or corrupt data due to a software fault that is traceable to one of its functional software components. "Correct" means that in the absence of external hardware or external software failures, the system performs all operations as defined in the Rolodex Tool formal specification.

The system must be robust in the face of hardware power failures in terms of data loss. "Robust" means that when computer power fails or is interrupted, the Rolodex Tool performs a `Save` operation on the currently open rolodex file, if the workspace is modified.

There are no specific requirements for security, privacy, safety, portability, modifiability, or extensibility.

3.2. Process

The system is to be developed using the software process employed by Gene Fisher in his software engineering classes at Cal Poly university. The process is currently described in lecture notes for the graduate course in software engineering.

3.3. Personnel

The development is to be conducted by a single individual (Gene Fisher) who is assumed to have the necessary knowledge and skills to complete the project. There are no specific requirements for the end users of the Rolodex Tool. Given its simple functionality, it is intended to be usable by end users with only basic computer literacy skills.

4. Developer Overview

See Section 4 of ˜/classes/440/handouts/final-report-overview

5. Requirements Rationale

We've chosen the "no effect" style of command execution in a number of cases where more feedback might be given to the users. *[Say a bit more here.]* The reason for this choice is simplicity.

We definitely opted for simplicity, as in exact match required for searching.

Appendix A. Users Manual

For organizational clarity, the users manual is presented in a separate stand-alone document. The manual contains no additional information from that presented in the functional requirements of Section 2. Rather, the manual presents Rolodex Tool functionality in a different form than the scenarios. The form is that of a software reference manual, which is terser and less tutorial than the scenario-style form of the requirements.

Appendix B. Help Content

This appendix specifies the complete content delivered for both quick help and detailed help.

B.1. Quick Help Messages

B.2. Detailed Help Content

Appendix C. UNIX Requirements

Appendix D. MacIntosh Requirements

References