Usability review report: A review of the user interface of the GFI E-mail Response Library.

Introduction

This is a review of the user interface (UI) of the GFI E-mail Response Library. While the system is powerful, problems in the UI may limit access to the information it contains. The report contains examples from the application with explanations of the problems I found and brief recommendations for fixing them.

Executive Summary

The desktop application is a great tool because it enables users to capture text and store it within file libraries but its User Interface has problems common to many applications. The stored text can then be accessed to be re-used in future communications.

This brief review includes problems in these categories:

- **Conceptual model:** The UI does not match the conceptual model that a typical user has of a knowledgebase catalog. Better organization of information and better visual design would help users understand search results and other screens.
- **Specifying searches:** Most computer users have problems specifying searches, so the catalog UI must make it as easy as possible.
- **Terminology:** The system uses many terms that users will not understand. Software should always employ the users’ terminology.
- **Flexibility and efficiency of use:** The system lacks accelerators which speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
- **Error prevention:** Even better than good error messages is a careful design which prevents a problem from occurring in the first place.
- **Consistency and Standards:** The users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions. The UI has too many visible options. Users need consistency to learn and use software.
- **User control and freedom:** Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue.

Types of Users considered in this report

While the library application is a powerful tool with a large database, I believe that the design does not reflect the needs of many of your users. This will lead to people not being able to take advantage of all that it offers.

The typical people that will use this UI are support and customer service people. Most of these users are novice to intermediate level which means that some of them are very comfortable with desktop/web software and others which are quite unsure of their abilities. Both will benefit from a more user friendly UI which will help them make use of this software on their own.

Since this application is being taken into consideration to be purchased by the general public, I believe that it can be a useful notepad tool for all types of users.

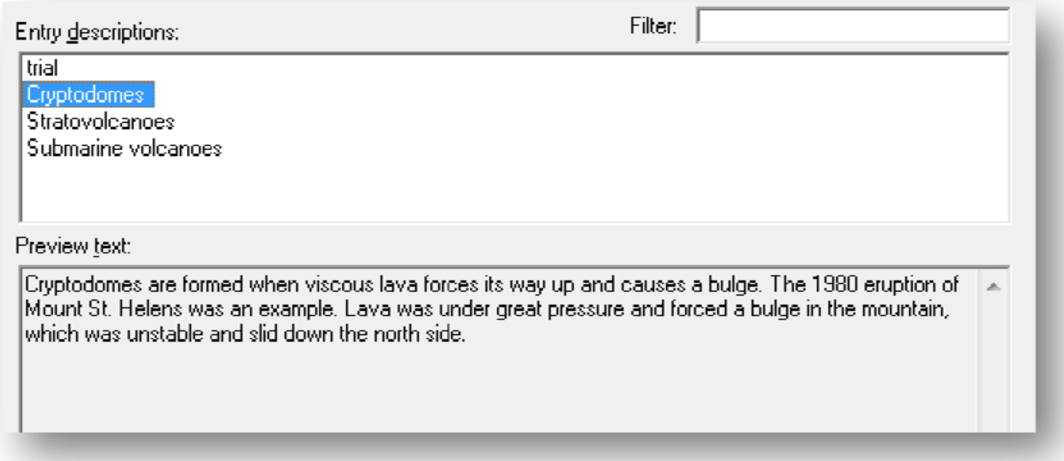
Problem: Using the conceptual model and specifying searches

Once the use copies some text by using copy or the F10 hotkey, he expects the system to give him a search result or the new entry screen.

The only visible search in this small application is the 'filter' label. If the user enters any word which is in any of the entries available, they will show in the 'preview text' text area.

When trying to search with an already available entry let's say 'Cryptodomes', the search does not return anything if the keyword 'Cryptodomes' is not available in the 'Preview text'

Also the 'Preview Text' here is useless and taking unnecessary space on the screen.



Cryptodomes are formed when viscous lava forces its way up and causes a bulge. The 1980 eruption of Mount St. Helens was an example. Lava was under great pressure and forced a bulge in the mountain, which was unstable and slid down the north side.

Example: The label for this is not descriptive of what it does, it should read something like ‘Search filter’ and the search screen should be separate to the ‘add/entry’ screen.

Recommendation 1: Limit search by

Provide ways to limit search by ‘Project’, which is a more suitable name than ‘library’ since the library is the whole database of the items.

Recommendation 2: Search availability

Many users look for search as soon as they use the application or webpage, so make sure that the search button is clearly visible and also an option to save a search for future use.

Recommendation 3: Simple navigation

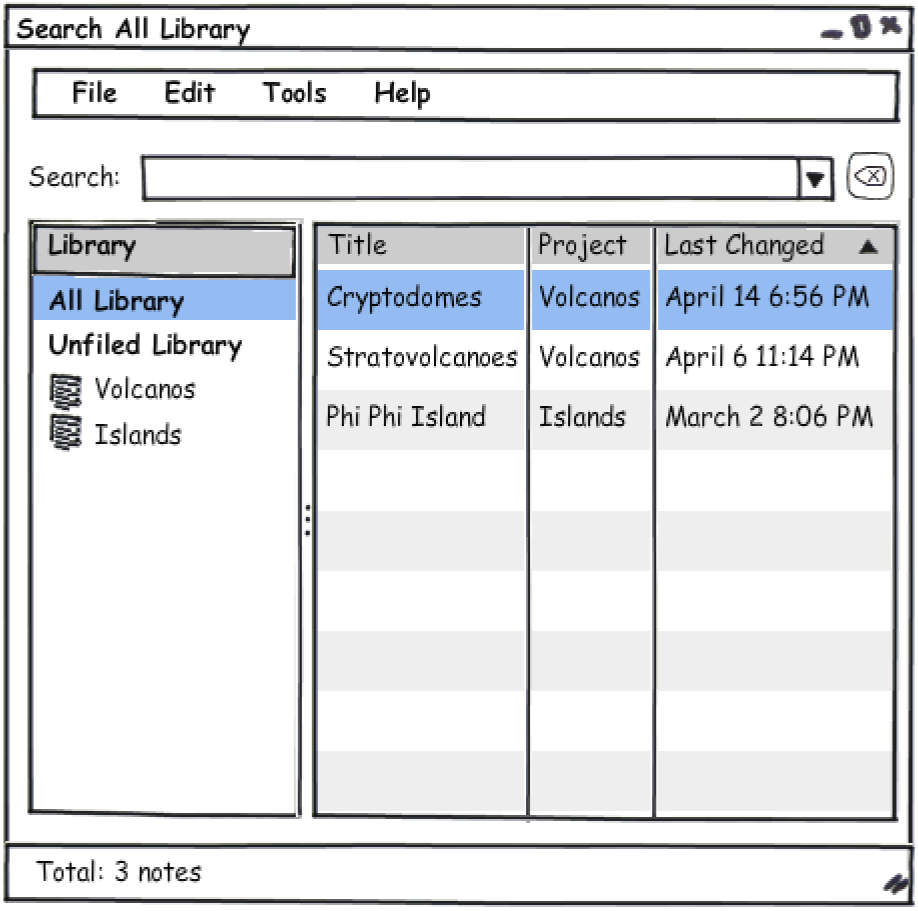
A simple navigation unlike currently in the system should be visible. Although the recommended system will have a menu with icons for easier user in other screens, this is would not be advisable here as it will take unnecessary space in this screen.

Recommendation 4: Search options

Provide at least two search options as seen in the sketch below. The first one is the text input area where the user can enter any keyboard and the system will list titles for that exact keyword match.

The other option is the sidebar ‘menu’ which will make it easier for the user if he just wants to search all ‘Titles’ under any of the available ‘Project’ items.

Sketch for design solution

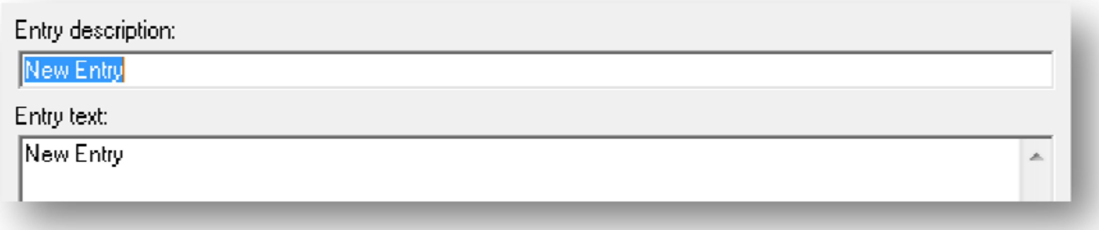


Title	Project	Last Changed ▲
Cryptodomes	Volcanos	April 14 6:56 PM
Stratovolcanoes	Volcanos	April 6 11:14 PM
Phi Phi Island	Islands	March 2 8:06 PM

Total: 3 notes

Problem: Unfamiliar Terminology

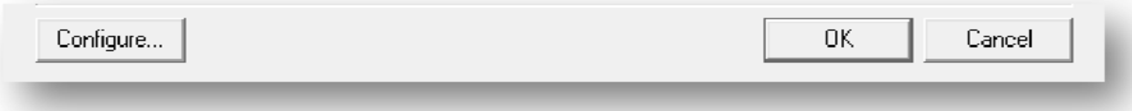
Library users will not understand library terminology and will be confused by it.



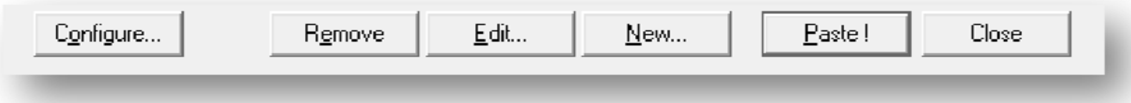
Entry description: New Entry

Entry text: New Entry

Example: The new entry screen gives the user the option to enter a 'description' and 'text' for the entry. The 'Enter description' is supposedly the title of this entry and the text is the proper entry. Most probably people do not know the difference between 'description' and 'text'.



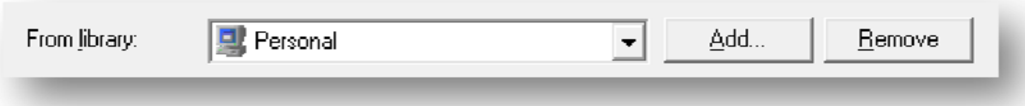
Example: The 'configure' button can be easily misunderstood and is not really needed in the screen area where the user is adding or editing the entries.



Also the above bar with several different buttons will definitely confuse the user. The 'configure' button which in reality is a preferences/tools section does not fit here.

No one will be able to predict the exact function of such buttons as the labels are not descriptive of their functions. The 'Paste' button for example is 'copy' to Email and does not really paste anything. Also after pressing that button a prompt message saying "Entry has been copied" or "Entry has been copied to clipboard" should be a helpful to the user instead of closing down the program itself.

Also the 'remove' button is positioned badly as the user can press that button by mistake when instead he is trying to 'edit' an entry. The 'remove' button should appear as part of the navigation (further down in the document) and will prompt the user with a screen before deleting the entry.



Example: The library bar in the top area of the screen can really be misunderstood. First time you try the software you do not get what the function of it is.

The term 'from library:' is quite confusing as the whole system is a library or database of all the entries. Here it is referring to a section of this library which I would call 'Project' or Section as an example.

Also the buttons beside the drop-down are not labeled something different from the ones at the bottom of the screen which relate to the entries.

Recommendation 5: Technical terms

Using technical terms (e.g., ‘configure’) will confuse users. All on-screen text should use users’ vocabulary, not technical wording.

Recommendation 6: Label naming

Labels should be clear enough so patrons immediately know how one command differs from another and know what they will get when they click on something.

Problem: Flexibility and efficiency of use

The system has one good accelerator which is the shortcut hotkeys which can be very useful for expert users.

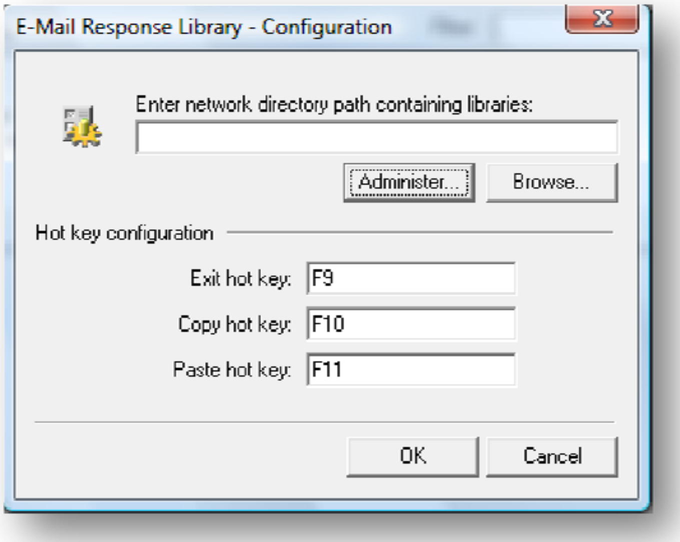
Example: Although hotkeys are useful, if this application is to be distributed to different users it should have a taskbar icon when it is running.

Example: The search as mentioned lacks functionality, a type-ahead auto completion would aid the frequent user when doing a search query.

Example: The user cannot change the name of the library (or a as I recommended the name ‘project’). One can only add new or just remove the project.

Example: The configure screen contains only one screen but has lots of configuration options that may confuse the user.

The ‘Administer’ option opens another screen which allows the user to move items from one library or project to another. However a missing option is to change the name of the project.



E-Mail Response Library - Configuration

Enter network directory path containing libraries:

Hot key configuration

Exit hot key: F9

Copy hot key: F10

Paste hot key: F11

Recommendation 7: Web application

This application should be a web application with an online database and not on the local computer.

Recommendation 8: Auto complete

Implement an auto-complete for the search. The system should ignore punctuation when comparing search parameters with database entries. Another possibility is to take punctuation into account only if the user specifies it in the query (e.g. in the preferences menu for expert users).

Recommendation 9: Add-ons

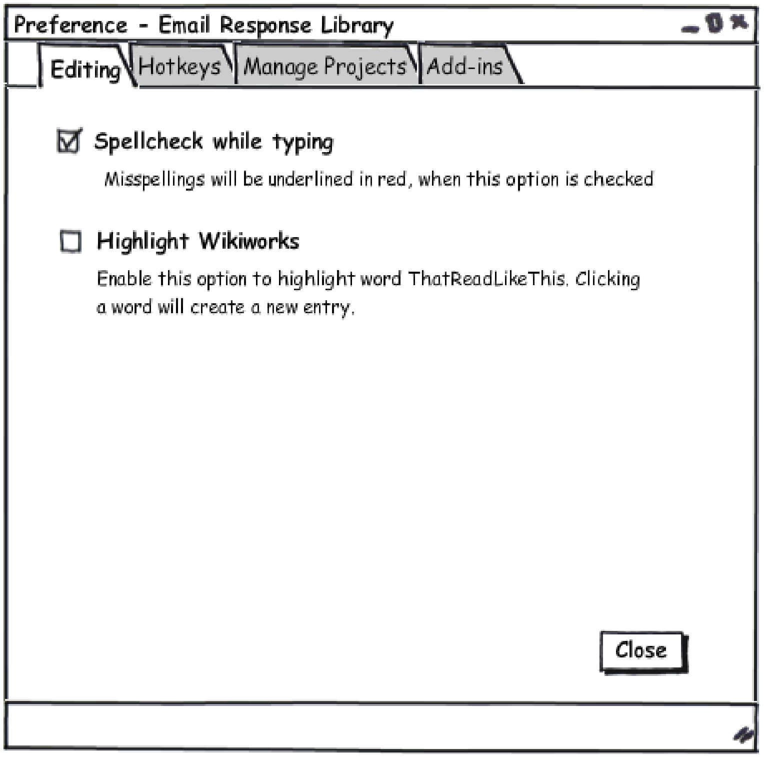
Add add-ons for example Microsoft Outlook. The functionality would be that when the user highlights text in an email the application will start the search there and then in one step.

Recommendation 10: Tools options

I would suggest changing the name of the ‘configure’ option to ‘tools’ as this is more an appropriate name for the user. Also the screen should be divided using tabs to allow for different options to be sort it out in a more cleaner way.

The ‘editing’ screen where there will be options related to text.

Sketch design option for the ‘tools’ screen.



☑ Spellcheck while typing

Misspellings will be underlined in red, when this option is checked

□ Highlight Wikiworks

Enable this option to highlight word ThatReadLikeThis. Clicking a word will create a new entry.

Recommendation 11: The ‘Manage Projects’ screen

One should explore the option of a drag and drop function with a prompt after the user has attempted to move something. Also I would make this option available according to user levels if the application is shared by a number of users so no data will be lost.

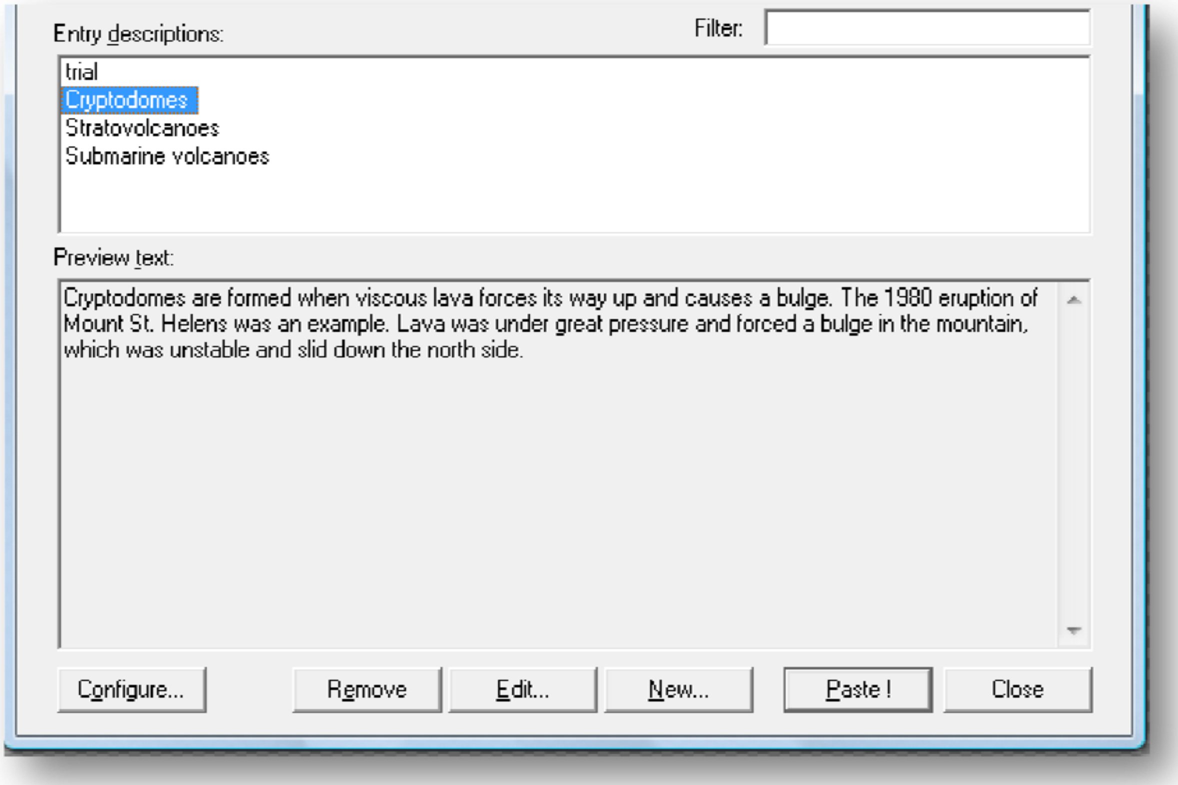
Problem: User control and freedom

Users are familiar with browser-based controls, including the Back button and Location field. However, using browser commands within a desktop application may result in data loss.

The application should include code that is aware and responsive to browser history.

One area that may be a better investment is working to ensure consistency in the application.

Example: When the user tries to delete an entry no prompt is given to the user before deleting the item and that item is lost forever.



trial
Cryptodomes
Stratovolcanoes
Submarine volcanoes

Cryptodomes are formed when viscous lava forces its way up and causes a bulge. The 1980 eruption of Mount St. Helens was an example. Lava was under great pressure and forced a bulge in the mountain, which was unstable and slid down the north side.

This can happen by mistake as the ‘remove’ button is near the ‘edit’ button and maybe the user just wanted to make changes to that entry.

Example: When the user presses the past button so that it is simply copied to the clipboard to be used for example in an email, the application just closes down without prompt. In this case if the user is new he will get confused and does not know what happened to the application.

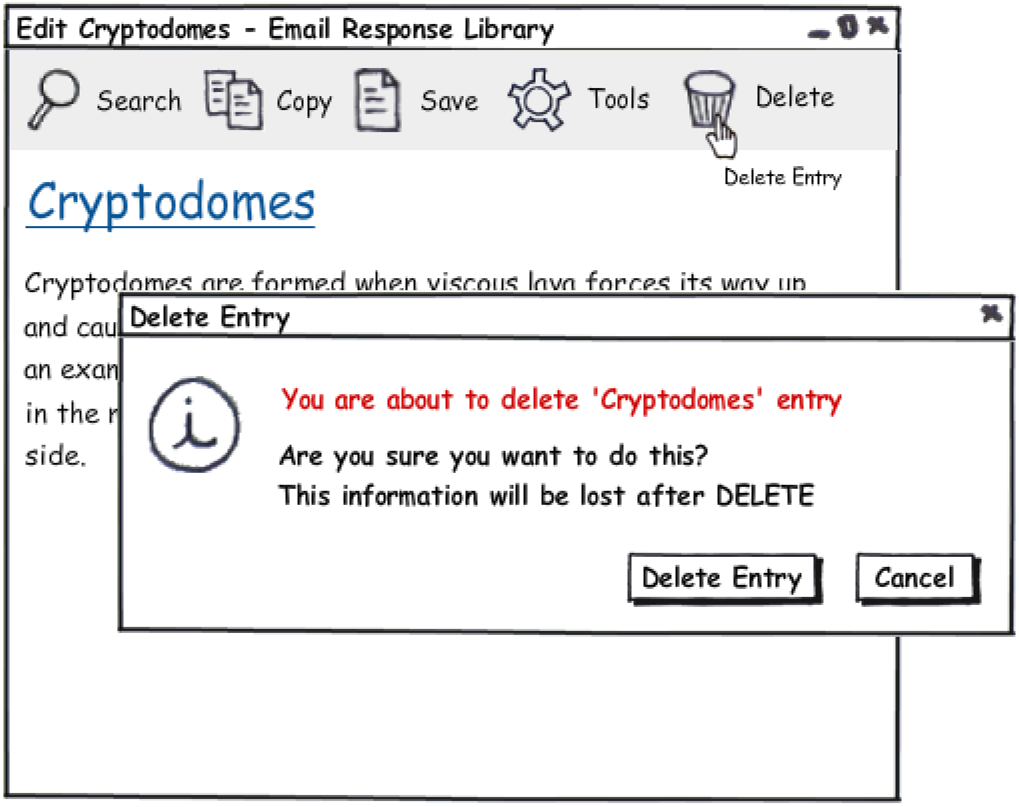
In another case if the user wants to look for another item the only available option is to use the F10 or F11 shortcut keys to have access to the application again.

Recommendation 12: Add user prompts

Prompts should be added when user tries to delete an item, however these should not be abused for every step the user does in the application.

Give the user short but important explanation of what he is about to do and an option to cancel and go back to the previous screen.

Sketch for design solution



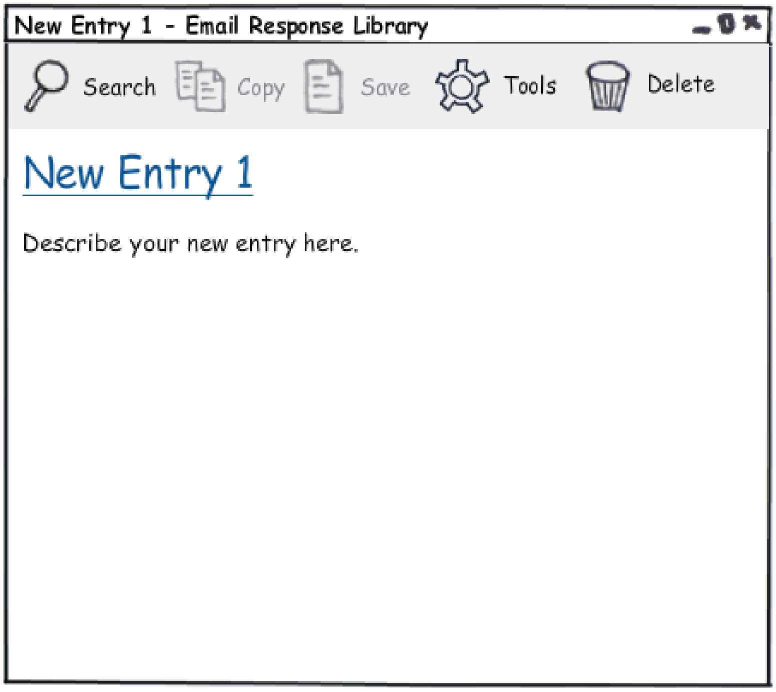
Cryptodomes are formed when viscous lava forces its way up
and cau
an exan
in the r
side.

You are about to delete 'Cryptodomes' entry

Are you sure you want to do this?
This information will be lost after DELETE

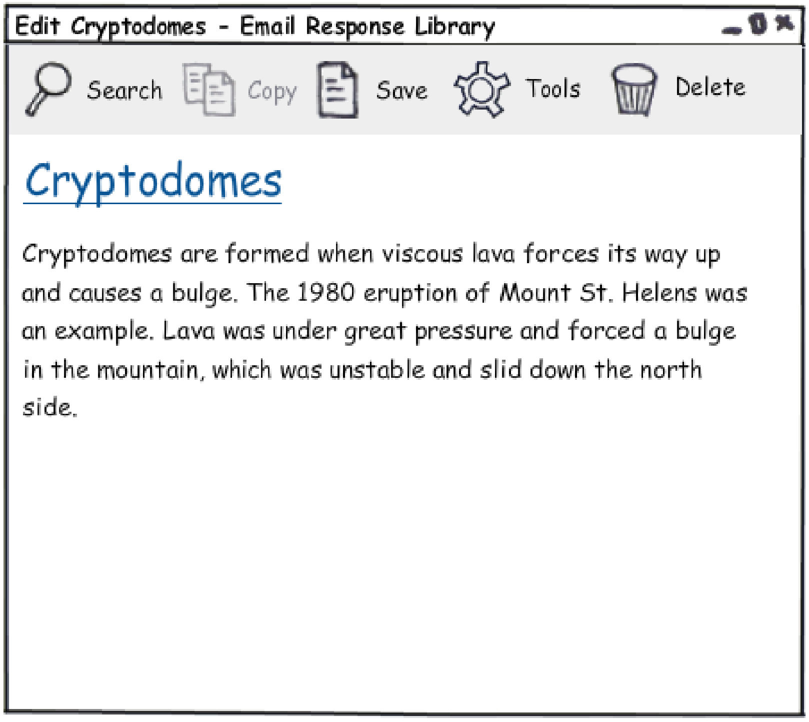
Other general recommendations:

The '**New Entry**' screen sketch design



Describe your new entry here.

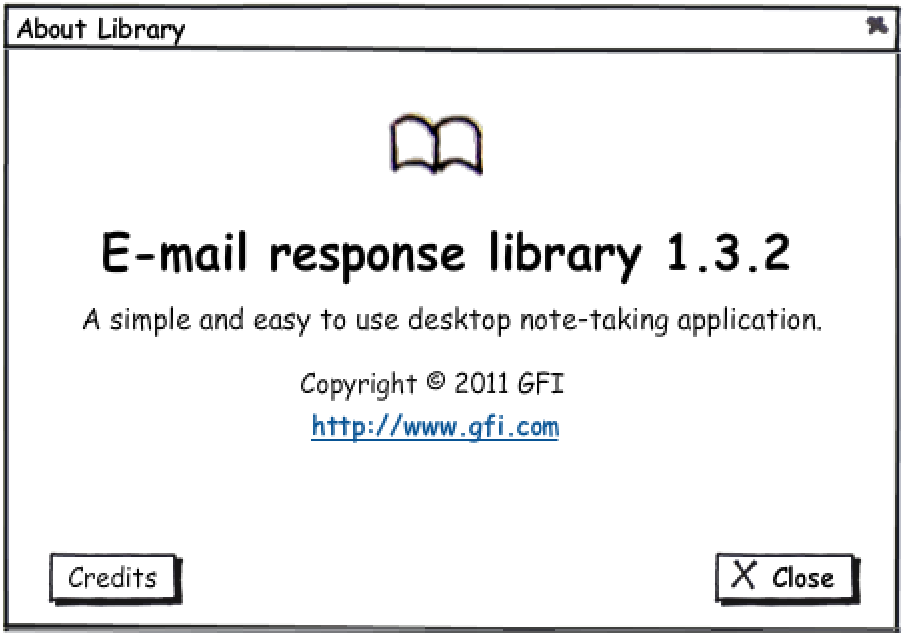
The **'Edit Entry'** screen sketch design



Cryptodomes

Cryptodomes are formed when viscous lava forces its way up and causes a bulge. The 1980 eruption of Mount St. Helens was an example. Lava was under great pressure and forced a bulge in the mountain, which was unstable and slid down the north side.

Provide a **splash screen** when the application has started.



E-mail response library 1.3.2

A simple and easy to use desktop note-taking application.

Copyright © 2011 GFI
http://www.gfi.com

Summary

This report was influenced using Jakob Nielsen’s heuristics and how one can apply these to desktop applications.

This report points out a number of problems in the user interface to GFI’s E-mail response library, which is a great resource to support staff and external users alike. There are many other problems, including layout and aesthetics design, the difficulty in using the limited search feature, and the difficulty in limiting a search to keyword search. These problems make it hard, perhaps impossible, for users to use the system and find what they need. This raises frustration and increases the workload of the support staff.

I believe that normal/beginner users are affected more than others. They are more likely to have few or no skills with computers, thereby making them not used to such applications. This hypothesis has been supported after looking into other systems available that can do same work. Also it would have been nice to have an informal discussion with users that have been using this application for some time so as to see what problems they have encountered.

This is admittedly a draft and not completed report, but I think I have mentioned most of the important problems in this software.

Typically each critical task has to be conducted many times, and each time a separate usability heuristic is evaluated against the task. Issues are noted, typically with screen shots captured and detailed information about the usability issue found.

Next steps include:

- Have a formal conversation with users of this application.
- Gathering data through surveys to collect large amounts of feedback.
- Running usability tests to get detailed feedback from users of the application.
- Using information from users and others to consult with GFI on ways to improve the UI to the catalog.