ADT: Site Search Version 1.0

Functional Specification

Author	Janet Tan
Version	1.0
Printed	2002-04-024:42 PM

`Document Revisions

Revisions on this document should be recorded in the table below:

	Date	Description	Author
1	18 February 2002	First Draft	Janet
2			
3			
4			
5			

1 Introduction

1.1 Purpose of this document

The purpose of this document is to specify the functional and system specification of the Site Search (Public) application. This document provides a road-map through which the implementation of the application was developed and, where applicable, will also contain information detailing technical design constraints and assumptions.

1.2 Scope of this document

The scope of this specification covers the functional specifications of the Site Search (Public) application. This specification does not extend to third party or infrastructure components (such as web server installation and/or configuration).

1.3 Overview

The basic requirements for the Site Search (Public) application include:

- Ability to search the keyword(s) from static html in the configurable server directory.
- Ability to display list of links to the static html on the server.
- Ability to display the selected static html.

1.4 Business Context

Site Search (Public) is a self-service application for software or customer support. It may be used wherever the need for a searchable body of knowledge is required. The generic version is open source for distribution.. Improvements to the base application will be made incrementally as bug reports come in and as suggestions are made. The generic Web Site Search (Public) will ultimately benefit both the company and our users.

2 General Description

2.1 Product Functions

The general functionality of the Site Search (Public) application is to allow users to access information online and to allow users in using keywords search to the static html.

2.2 General Constraints

The Site Search (Public) is intended to be just a simple search for static html on the server site.

The application is written in Perl and configured using the standard general distribution specifications

Bold Params indicate Section Headings. *Italic Params* are objects.
Indented Params are configurable variables.
Grayed out or missing items are not required or not available in the default configuration.

Param	Default configuration	Notes
Language	Perl	
Search Directory Setup		
Search Directories	User has to fill in the directories	
View Setup		Views to use, display, and defaults
Valid Views	CSSView, SearchKeywordsView	The program cannot display views that are not available here.
View Display Params		Defaults for Views
Application Logo	logo.gif	
Application Logo Height	40	
Application Logo Width	353	
Application Logo ALT	WebDB Demo	
Color for Even Rows	E5E5E5	
Color for Odd Rows	FFFFFF	
Display Fields		
Document Root URL	/	
Field Name Mappings		
Image Root URL	/Images/Extropia	
Link Target	_self	
Row Color Rules	\@ROW_COLOR_RULES	
Script Display Name	Site_search	
Script Name	$CGI->script_name()	
Selected Display Fields		
Sort Fields		
Table BG Color 1	6699CC	
Table BG Color 2	E5E5E5	
Table BG Font Color 1	BLACK	
Table BG Font Color 2	BLACK	
Action/Workflow Setup		
Action Handler List	Default::DisplayCSSViewAction Search::SearchKeywordsAction Search::DefaultAction	
Action Handler Action Params		
Action Handler List	\@ACTION_HANDLER_LIST	
Search View	SearchKeywordsView	
Search Directories	\@SEARCH_DIRECTORIES	
Application Sub Menu View Name	SiteSearch_ApplicationSubMenuView	
CGI Object	$CGI	
CSS View URL	$CGI->script_name() . "?display_css_view=on"	URL location of the CSS template. Should be changed to a location outside the cgi-bin for speed.
CSS View Name	CSSView	You can rename HTMLTemplates/CSSView.ttml to CSSView.css or

		whatever is preferred.
Templates Cache Directory	$TEMPLATES_CACHE_DIRECTORY	“../../Datafiles/TemplatesCache”
Valid Views	\@VALID_VIEWS	
View Display Params	\@VIEW_DISPLAY_PARAMS	
View Loader	$VIEW_LOADER	
Page Top View	PageTopView	Header template
Page Bottom View	PageBottomView	Footer template
HTML Document Root	$documents_to_be_searched_path	You can define the documents root path to be search or you can extract from $ENV{'DOCUMENT_ROOT'}

Server URL	$http_host	my $server_name
		= $CGI->server_name;
		my $server_port
		= $CGI->server_port;
		
		if($server_port != 80){
		$server_name =
		$server_name . ":" .
		$server_port;
		}
		
		my $http_host
		= "http://". $server_name;

First Record On Page	$CGI->param('first_record_to_display') || 0	Get the first displayed record on the page
Last Record On Page	$CGI->param('first_record_to_display') || 0	Get the last displayed record on the page
Max Records Per Page	$CGI->param('records_per_page') || 25	Get the maximum record that can be displayed on a page
Require Details Footer View	1	1 => On. It will display only records in set of maximum records per page. 0 => Off. It will display all the records not restricting to the maximum records per page.

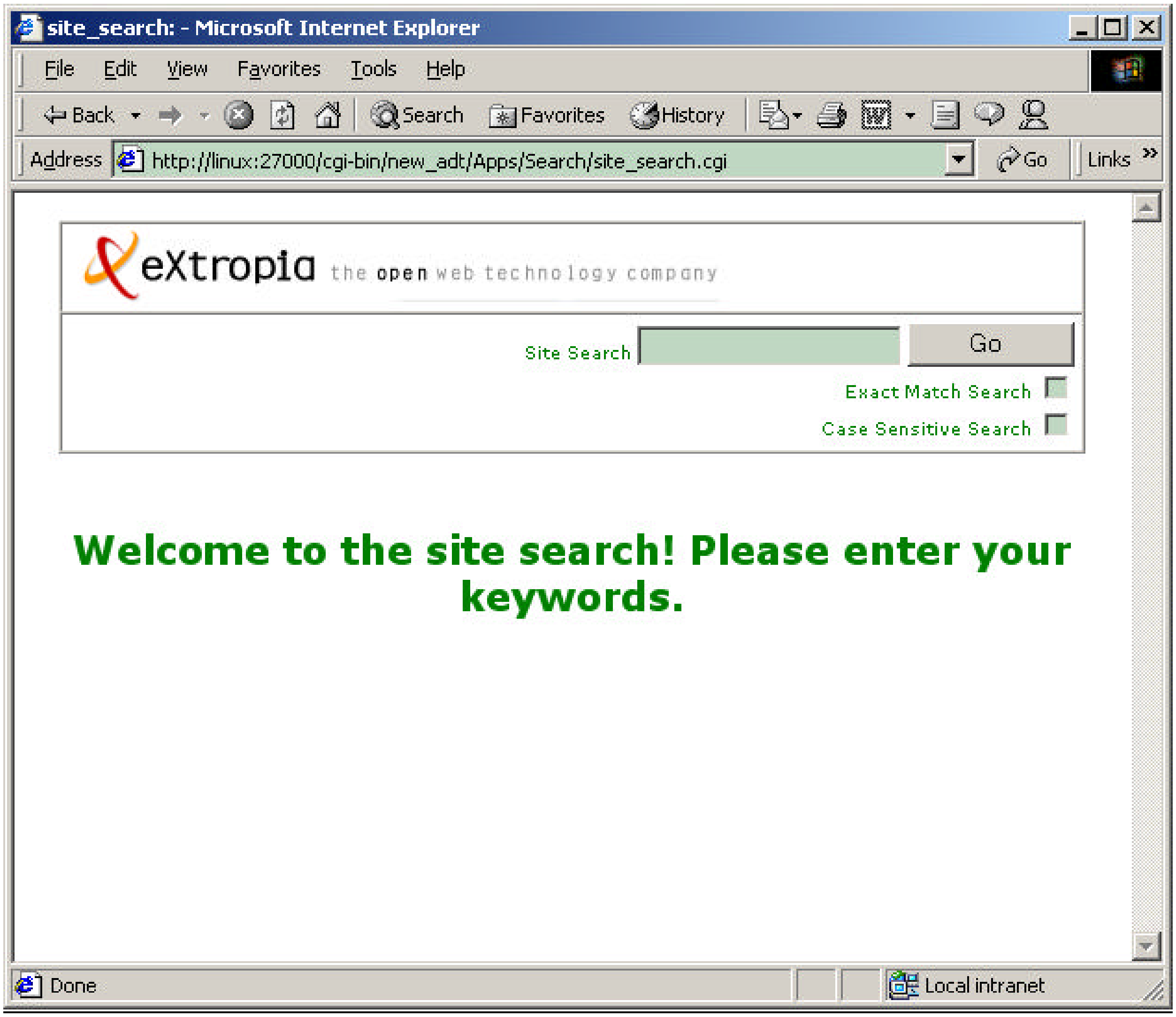
3 Functional Requirements

3.1 Site Search (Public) Application

3.1.1 Description

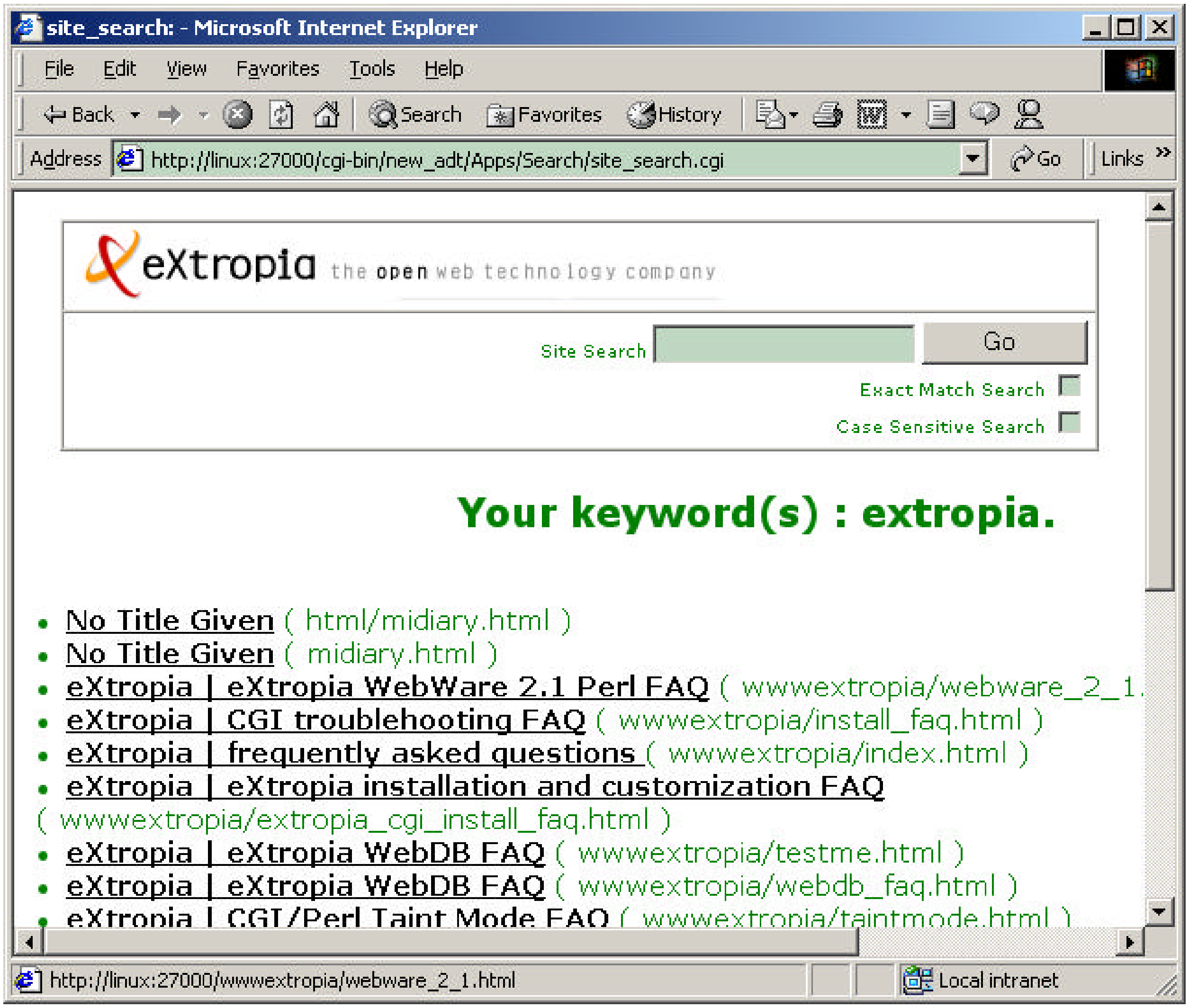
The following definitions only cover search view.

No authentication is required.



http://linux:27000/cgi-bin/new_adt/Apps/Search/site_search.cgi

Welcome to the site search! Please enter your keywords.



http://linux:27000/cgi-bin/new_adt/Apps/Search/site_search.cgi

Site Search

Exact Match Search □

Case Sensitive Search □

Your keyword(s) : extropia.

- No Title Given (html/midiary.html)
- No Title Given (midiary.html)
- eXtropia | eXtropia WebWare 2.1 Perl FAQ (wwwextropia/webware_2_1.
- eXtropia | CGI troublehooting FAQ (wwwextropia/install_faq.html)
- eXtropia | frequently asked questions (wwwextropia/index.html)
- eXtropia | eXtropia installation and customization FAQ (wwwextropia/extropia_cgi_install_faq.html)
- eXtropia | eXtropia WebDB FAQ (wwwextropia/testme.html)
- eXtropia | eXtropia WebDB FAQ (wwwextropia/webdb_faq.html)
- eXtropia | CGI/Perl Taint Mode FAQ (wwwextropia/taintmode.html)

http://linux:27000/wwwextropia/webware_2_1.html

4 Search Design

4.1 Description

The search is solely on the server site. File::Find will be used to do the search. File::Find – will traverse a file tree . (Refer to perldoc File::Find for more information)

SYNOPSIS

```
use File::Find;
        find(\&wanted, @search_directories);
        sub wanted { ... }
```

The wanted() function will does whatever searching of the keywords or verification.

Basic tasks in wanted()function for the site search application are as below:

1) Check if file in search_directories ends with .html extension.
2) Open the .html file.
3) While not end of file.
4) Get the title of the file.
5) Check if all the keywords are found.
6) Close the file.
7) If all the keywords are found, add the file link to html_output.

4.2 Communications Interfaces

5 Other non-functional attributes

5.1 Reliability

5.2 Maintainability

Requires little maintenance.

5.3 Portability

The application may be used on any architecture where Perl is used.

5.4 Extensibility

5.5 Reusability

Site Search (Public) is highly reusable. The application can be modified for just about anything!

5.6 Serviceability