

ADT: Guest Book Version 1.0

Requirements Specification

Author	Janet
Version	1.0
Printed	2001-10-22 12:52 PM

Document Revisions

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

	Date	Description	Author
1	10 September 2001	First Draft	Janet
2	10 September 2001	First Edit	Josh
3	12 October 2001	Added Screenshots, Views, Params	Josh
4			

Table of Contents

1 Introduction	4	
1.1 Purpose of this document		4
1.2 Scope of this document		4
1.3 Overview		4
1.4 Business Context		4
2 General Description	4	
2.1 Product Functions		4
2.2 General Constraints		4
3 Functional Requirements	8	
3.1 Guestbook Application		8
3.1.1 <i>Description</i>		8
3.1.2 <i>Risks and Dependencies</i>		11
4 Database Design	11	
4.1 Description		11
4.2 Communications Interfaces		11
5 Other non-functional attributes	11	
5.1 Reliability		11
5.2 Maintainability		11
5.3 Portability		11
5.4 Extensibility		11
5.5 Reusability		11
5.6 Serviceability		12

1 Introduction

1.1 Purpose of this document

The purpose of this document is to specify the functional and system specification of the Guestbook application. This document provides a road-map through which the implementation of the application will be 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 Guestbook application. This specification does not extend to third party or infrastructure components.

1.3 Overview

The basic requirements summary built-in for the Guestbook application:

- To allow users to write their comments in the Guestbook and view by the public.
- To be able to perform a keyword search.

1.4 Business Context

Guestbook is an application designed for users to send their comments to the website. The generic version is open source for distribution. Anyone may take advantage of the powerful eXtropa Perl ADT. Improvements will be made incrementally as bug reports come in and suggestions are made. The generic Guestbook will ultimately benefit both the company and our users.

2 General Description

2.1 Product Functions

The general functionality of the Guestbook application is to allow users to send their comments and view by public. The application generates a dynamic page depending on the entered information and the setup preferences.

2.2 General Constraints

The application will be written in Perl and the data files are stored in a common flatfile system (ASCII File). This will slow down in terms of searching for information if there is a presence of huge database.

The application is also able to interface with relational databases should the administrator desire it.

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	
Data Handler Setup		How data is handled
<i>Add Form DHM Config</i>		
Type	CGI	
CGI Object	\$CGI	
Data Handlers	Email, Exists, HTML, String	
Field Mappings	Fname, lname, email, location, comments	
Datasource Setup		Configure the database and fields displayed
Field Names	Fname, lname, email, location, comments	Corresponds to default extropia database
<i>Sample Default Widget Definitions</i>		Configure field types for TTML
Display Name	Country	Name that appears in the browser
Type	Textfield, textarea	
Name	Fname	Internal field name
Size	30	Display size in characters
Maxlength	80	Maximum number of characters permitted
Rows	6	Initial height of textarea
Cols	30	Initial width of textarea
Wrap	VIRTUAL	Wrapping of text in textarea
<i>Basic Datasource Config Params</i>		Configure datasource
Type	File	
File	\$APP_DATAFILES_DIRECTORY/\$APP_NAME.dat	
Field Delimiter	(pipe)	
Comment Prefix	#	Ignore lines starting with #
Create File if None Exists	On	
Field Names	\@DATASOURCE_FIELD_NAMES	
Key Fields	['record_id']	Not required if key is not used
Field Types	Record_id => 'Autoincrement'	Specify a specific kind of field
Mailer Setup		Required for regular email. Not the same as Auth Mailer
<i>Mail Config Params</i>		
Type	Sendmail	
<i>Email Display Fields</i>	Subject, abstract, full_text	Fields that will be emailed
<i>Add Event Mail Send</i>		Tells user of additions
From	you@yourdomain.com	
To	you@yourdomain.com	

Reply To	you@yourdomain.com	
Subject	Guestbook Addition	
Logging Setup		
Type	File	
Log File	\$APP_DATAFILES_DIRECTORY/\$APP_NAME.log	
Log Entry Suffix	' ' . _generateEnvVarsString() . ' ',	
Log Entry Prefix	'Guestbook '	
View Setup		
<i>Valid Views</i>	CSSView, AddRecordView, BasicDataView, DetailsRecordView, AddAcknowledgementView, AddRecordConfirmationView,	Views to use, display, and defaults The program cannot display views that are not available here.
<i>View Display Params</i>		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	Fname, lname, email, location, comments	
Table BG Color 1	6699CC	
Table BG Color 2	E5E5E5	
Table BG Font Color1	BLACK	
Table BG Font Color2	BLACK	
Filter Setup		
<i>HTMLIZE Filter</i>		
Type	HTMLize	
Convert Double Linebreak to P	On	
Convert Linebreak to BR	On	
<i>Charset Filter</i>		
Type	CharSet	
<i>Embed Filter</i>		
Type	Embed	
Enable	Off	
Action/Workflow Setup		
<i>Action Handler List</i>	Defaults	User should not need to modify
<i>Action Handler Action Params</i>		
Action Handler List	\@ACTION_HANDLER_LIST	
Add Acknowledgment View Name	AddAcknowledgementView	
Add Email Body View	AddEventEmailView	
Add Form View Name	AddRecordView	
Allow Additions Flag	On	
Application Sub Menu View Name	ApplicationSubMenuView	
Options Form View Name	OptionsView	
Auth Manager Config Params	\@AUTH_MANAGER_CONFIG_PARAMS	

Add Record Confirmation View Name	AddRecordConfirmationView	
Basic Data View Name	BasicDataView	
CGI Object	\$CGI	
CSS View URL	\$CSS_VIEW_URL	For speed, change CSS URL to a location outside the CGI directory.
CSS View Name	CSSView	
Default Sort Field 1	Title	
Default Sort Field 2	abstract	
Datasource Config Params	\@DATASOURCE_CONFIG_PARAMS	
Delete Acknowledgment View	DeleteAcknowledgementView	
Delete Record Confirm View	DeleteRecordConfirmationView	
Records Per Page Options	[5, 10, 25, 50, 100]	Remove or change numbers as desired. This appears under Options.
Max Records Per Page	\$CGI->param('records_per_page') 5	
Sort Field 1	\$CGI->param('sort_field1') 'category'	Sort by which field?
Sort Field 2	\$CGI->param('sort_field2') 'fname'	Sort by which field?
Sort Direction	\$CGI->param('sort_direction') 'ASC'	ASC=Ascending, DSC=Descending
Delete Form View Name	BasicDataView	
Delete Email Body View	DeleteEventEmailView	
Details View Name	DetailsRecordView	
Data Handler Manager Config Param	\@DATA_HANDLER_MANAGER_CONFIG_PARAMS	
Display Acknowledgement On Add	On	
Display Acknowledgement On Delete	On	
Display Acknowledgement On Modify	On	
Display Confirmation On Add	On	
Display Confirmation On Delete	On	
Display Confirmation On Modify	On	
Enable Sorting	On	
Hidden Admin Fields View	HiddenAdminFieldsView	
Input Widget Definitions	\@INPUT_WIDGET_DEFINITIONS	
Key Field	record_id	
Logoff View	LogoffView	

URL Encoded Admin Fields View	URLEncodedAdminFieldsView	
Log Config Params	ModifyAcknowledgementView	
Modify Acknowledgment View	ModifyRecordConfirmationView	
Mail Config Params	\@MAIL_CONFIG_PARAMS	
Modify Form View	ModifyRecordView	
Modify Email Body View	ModifyEventEmailView	
Send Email on Add	On	
Templates Cache Directory	\$TEMPLATES_CACHE_DIRECTORY	
Valid Views	\@VALID_VIEWS	
View Display Params	\@VIEW_DISPLAY_PARAMS	
View Filters Config Params	\@VIEW_FILTERS_CONFIG_PARAMS	
View Loader	\$VIEW_LOADER	
Simple Search String	\$CGI->param('simple_search_string') ""	
First Record on Page	\$CGI->param('first_record_to_display') 0	
Last Record on Page	\$CGI->param('first_record_to_display') "0"	
Page Top View	PageTopView	
Page Bottom View	PageBottomView	
Action Handler Plugins	\%ACTION_HANDLER_PLUGINS,	
Basic Input Widget Display COLSPAN	2	

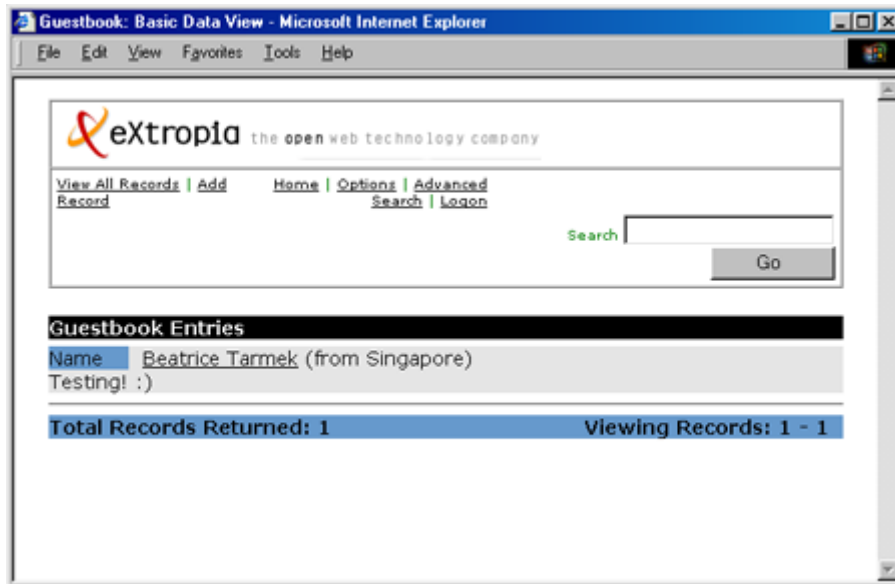
3 Functional Requirements

3.1 Guestbook Application

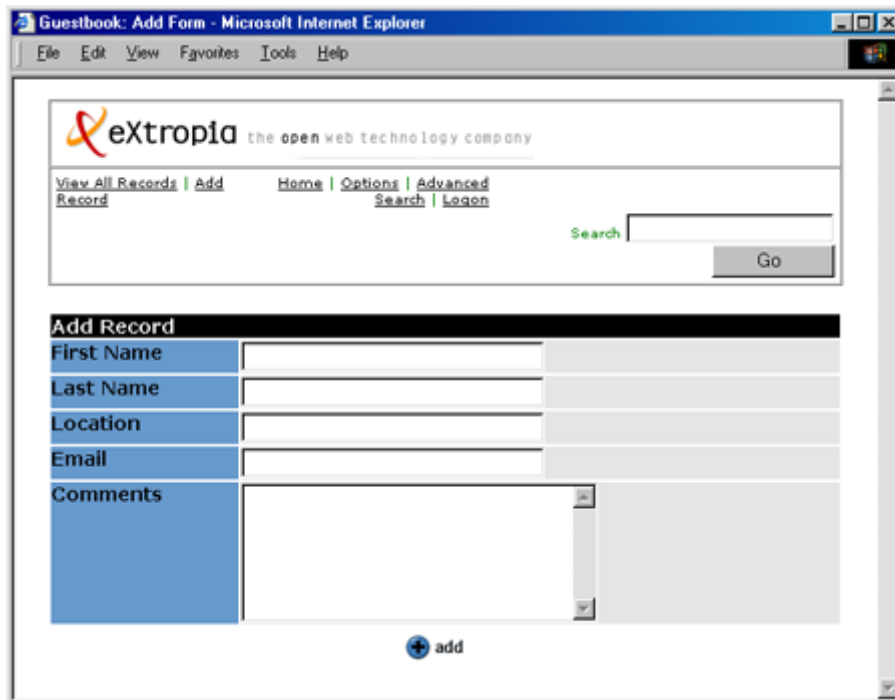
3.1.1 Description

Any user may add a record or examine the records.

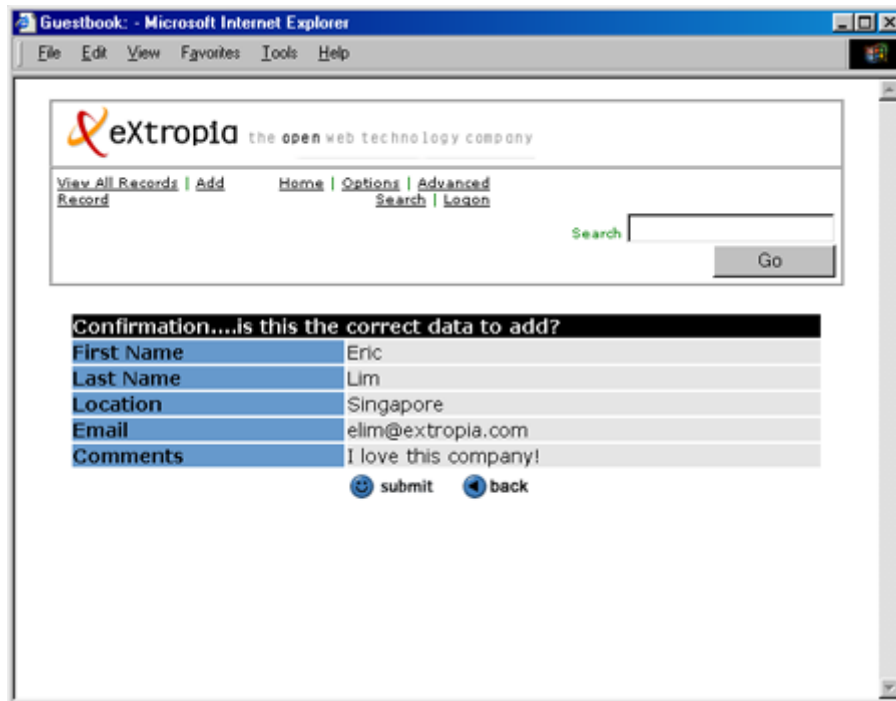
- **Home View** displays the current records. The user may Add, Show All, go Home, and perform a Keyword Search. The other options are not available.



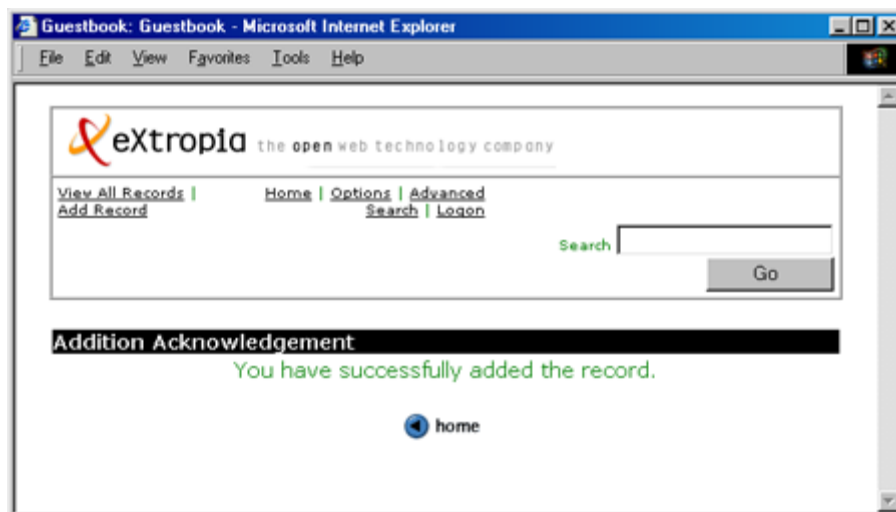
- **Add Record** permits the user to Add an entry with First Name, Last Name, Location, Email, and Comments.



- **Confirmation Page** confirms the information the user entered is accurate.



- **Acknowledgement Page** confirms the addition and allows the user to return Home.



- **Basic Search** will use the Home View but only display records that match the keyword(s). No records are displayed if none match.

3.1.2 Risks and Dependencies

4 Database Design

4.1 Description

The data of the application will be stored in a flat file. Each data fields of a record will be separated by '|'.

Field Name	Description
record_id	Record value (Auto Incremental Value)
fname	First Name
lname	Last Name
location	Location
email	Email
comments	Comments
reviewed_by_admin	Reviewed By Administrator
date_time_posted	Date and Time the record is posted.

4.2 Communications Interfaces

5 Other non-functional attributes

5.1 Reliability

Highly reliable in a properly secured environment.

5.2 Maintainability

Requires little maintenance. Owners will want to clear out a large flatfile now and then as well as ensure their installation is secure from hackers. Users of relational databases will probably not need to clear out their database.

5.3 Portability

5.4 Extensibility

Information entered is portable to any relational database. The program can easily be switched to a new database with just a few lines of code. The program relies on the eXtropa ADT that divorces form and function.

5.5 Reusability

Guestbook is highly reusable. The application can be modified for just about anything!

5.6 Serviceability