

# ADT: Inventory Manager Version 1.0

Functional Specification

<b>Author</b>	Jason
<b>Version</b>	1.0
<b>Printed</b>	2001-10-22 12:58 PM

## Document Revisions

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

	<b>Date</b>	<b>Description</b>	<b>Author</b>
1	10 September 2001	First Draft	Jason
2	10 September 2001	First Edit	Josh
3	12 October 2001	Added Screenshots, Params, Views, Editing	Josh
4			

<b>1 Introduction</b>	<b>4</b>	
1.1 Purpose of this document		4
1.2 Scope of this document		4
1.3 Overview		4
1.4 Business Context		4
<b>2 General Description</b>	<b>5</b>	
2.1 Product Functions		5
2.2 General Constraints		5
<b>3 Functional Requirements</b>	<b>11</b>	
3.1 Inventory Manager Application		11
3.1.1 <i>Description</i> .....		11
3.1.2 <i>Risks and Dependencies</i> .....		19
<b>4 Database Design</b>	<b>19</b>	
4.1 Description		19
4.2 Communications Interfaces		19
<b>5 Other non-functional attributes</b>	<b>20</b>	
5.1 Reliability		20
5.2 Maintainability		20
5.3 Portability		20
5.4 Extensibility		20
5.5 Reusability		20
5.6 Serviceability		20

# 1 Introduction

## 1.1 Purpose of this document

The purpose of this document is to specify the functional and system specification of the Inventory Manager 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 Web Inventory Manager application. This specification does not extend to third party or infrastructure components (such as a web server installation and/or configuration).

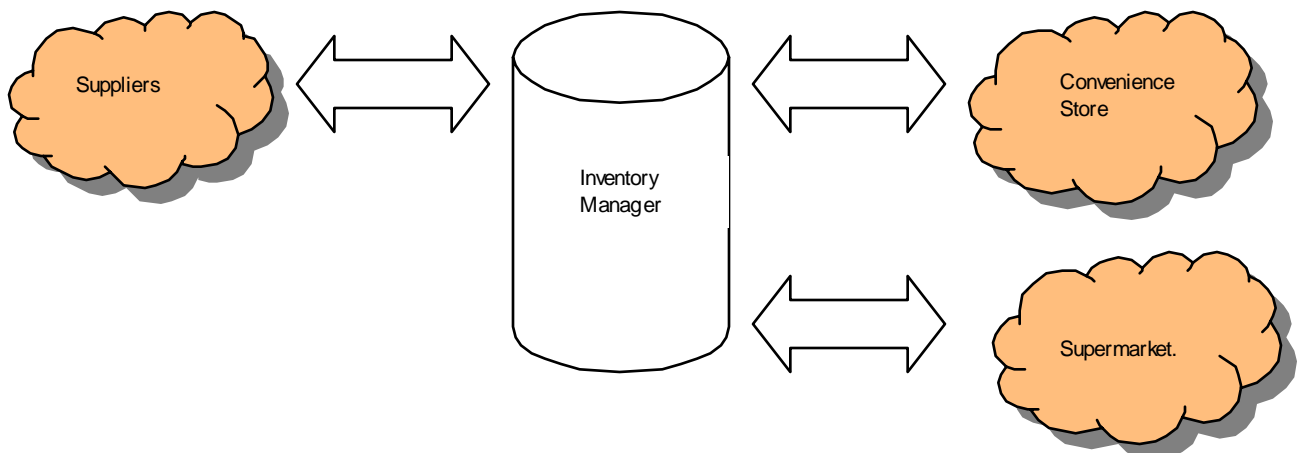
## 1.3 Overview

The basic requirements for Inventory Manager are:

- To let users maintain the information on inventories via the Web.
- To restrict usage to authorized users.
- In our default distribution, we make use of an arbitrary Audio records store to keep track of the quantity of the records available and the price of each piece.

## 1.4 Business Context

Inventory Manager is one of the groupware application designed for multiple users over a wide area. The generic version is open source for distribution. Anyone may take advantage of the powerful eXtropa Perl ADT. Improvements will be made incrementally. The generic Web Inventory Manager will ultimately benefit both the company and our users. This application covers most of the online inventory status that any store would require.



## 2 General Description

### 2.1 Product Functions

The general functionality of the Inventory Manager application is to allow users to display supply information in a single location online. The application generates a dynamic page depending on the entered information and the setup preferences. Inventory Manager permits additions and deletions of database records.

### 2.2 General Constraints

The application is written for restricted posting. A definite login is required for viewing information and to add or modify any information including new record.

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
<b>Language</b>	Perl	
<b>Session Setup</b>		
Type	File	Note that flatfile data sources are slow. We recommend upgrading to a database based data source
Max Modify Time	1 hour	
Fatal Timeout	Off	
Fatal if session not found	Off	
<i>Session Manager Setup</i>		
Type	FormVar	
Cgi Object	\$CGI	
Session Params	Session Config Params	
<b>Authentication Setup</b>		For the datasource
Type	File	
Field Delimeter	(pipe)	
Create a file if none exists	On	
Field Names	From Auth User Datasource	
File	Scriptname.users.dat	Path is set at start of script
Default Authentication Fields	Username, password, groups, firstname, lastname, email	
<b>Authentication Manager</b>		How requests are

<b>Setup</b>		processed
Color for even rows	E5E5E5	
Color for Odd Rows	FFFFFF	
CSS View URL	\$CSS_VIEW_URL	
Application Logo	logo.gif	
Application Logo Height	40	
Application Logo Width	353	
Application Logo ALT	WebDB Demo	
HTTP Header Params	[-EXPIRES => '-1d']	
Document Root URL	'/' or web root directory	
Image Root URL	/Images/Extropia	
Script Display Name	'InventoryManager'	
Script Name	\$CGI->script_name()	
Page Top View	PageTopView	
Page Bottom View	PageBottomView	
Link Target	'_self' or current window	
<i>User_Fields</i>	Auth_username, auth_password, auth_groups, auth_firstname, auth_lastname, auth_email	
<i>Mail Params</i>		
Type	Sendmail	Default on most systems
<i>User Mail Send Params</i>		
From	you@yourdomain.com	
Subject	Password Generated	
<i>Admin Mail Send Params</i>		
From	you@yourdomain.com	
To	you@yourdomain.com	
Subject	Registration Notification	
<i>Auth Manager Config Params</i>		
Type	CGI	
Allow Registration	On	
Allow User Search	On	Allow anyone to check for an old account.
Generate Password	Off	
Default Groups	Normal	
Email Registration to Admin	Off	
Display Registration After Failure	On	
<b>Data Handler Setup</b>		How data is handled
<i>Add Form DHM Config</i>		
Type	CGI	
CGI Object	\$CGI	
Data Handlers	Exists, HTML, String	
Field Mappings	Artist, album_title, price, distributor, quantity	
<b>Datasource Setup</b>		Configure the database and fields displayed
Field Names	record_id, artist, album_title, price, distributor, quantity, username_of_poster, group_of_poster, date_time_posted,	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
<b>Basic Datasource Config Params</b>		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' , quantity => 'Number'	Specify a specific kind of field
<b>Mailer Setup</b>		Required for regular email. Not the same as Auth Mailer
<b>Mail Config Params</b>		
Type	Sendmail	
<b>Email Display Fields</b>	Artist, album_title, price, distributor, quantity	Emails the fields
<b>Delete Event Mail Send</b>		Warns Someone of Deletions to the database
From	you@yourdomain.com	
To	you@yourdomain.com	
Reply To	you@yourdomain.com	
Subject	Inventory Delete	
<b>Add Event Mail Send</b>		Tells user of additions
From	you@yourdomain.com	
To	you@yourdomain.com	
Reply To	you@yourdomain.com	
Subject	Inventory Addition	
<b>Modify Event Mail Send</b>		Tells user of modifications to records
From	you@yourdomain.com	
To	you@yourdomain.com	
Reply To	you@yourdomain.com	
Subject	Inventory Modification	
<b>Logging Setup</b>		
Type	File	
Log File	\$APP_DATAFILES_DIRECTORY/\$APP_NAME.log	
Log Entry Suffix	' ' . _generateEnvVarsString() . ' ',	

Log Entry Prefix	'InventoryManager '	
<b>View Setup</b>		Views to use, display, and defaults
<i>Valid Views</i>	CSSView, AddRecordView, BasicDataView, DetailsRecordView, AddAcknowledgementView, AddRecordConfirmationView, DeleteRecordConfirmationView, DeleteAcknowledgementView, ModifyAcknowledgementView, ModifyRecordConfirmationView, ModifyRecordView, PowerSearchFormView, SessionTimeoutErrorView, LogoffView, OptionsView,	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	Artist, album_title, price, distributor, quantity	
Table BG Color 1	6699CC	
Table BG Color 2	E5E5E5	
Table BG Font Color1	BLACK	
Table BG Font Color2	BLACK	
<b>Filter Setup</b>		
<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	
<b>Action/Workflow Setup</b>		
<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	
Allow Deletions Flag	On	
Allow Duplicate Entries	Off	
Allow Username Fields to be Searched	On	
Allow Modifications	On	



Flag		
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	
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	
Default Sort Field 1	artist	
Default Sort Field 2	album_title	
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	

On Modify		
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	
Power Search View	PowerSearchFormView	
Require Auth to Search	Off	Is authorization required for the action?
Require Auth for Add	On	Is authorization required for the action?
Require Auth for Modify	On	Is authorization required for the action?
Require Auth for Delete	On	Is authorization required for the action?
Require Auth for Details	Off	Is authorization required for the action?
Require Matching Username for Modify	Off	Is a matching group or username required for the action on that record?
Require Matching Group for Modify	Off	Is a matching group or username required for the action on that record?
Require Matching Username for Delete	Off	Is a matching group or username required for the action on that record?
Require Matching Group for Delete	Off	Is a matching group or username required for the action on that record?
Require Matching Username for Search	Off	Is a matching group or username required for the action on that record?
Require Matching Group for Search	Off	Is a matching group or username required for the action on that record?
Send Email on Delete	Off	
Send Email on Modify	Off	
Send Email on Add	Off	
Session Object	\$SESSION	
Session Timeout View	SessionTimeoutErrorView	
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 Inventory Manager Application

#### 3.1.1 Description

The following definitions only cover the Default settings and views.

##### **Authentication Definition Views**

The application will allow any user read-only rights to view the records registered in the application. Only registered users however, may add, modify, and delete records. There is no capacity for multiple group definitions, thus the default group is "Normal" and cannot be changed.

Default Settings Permit:

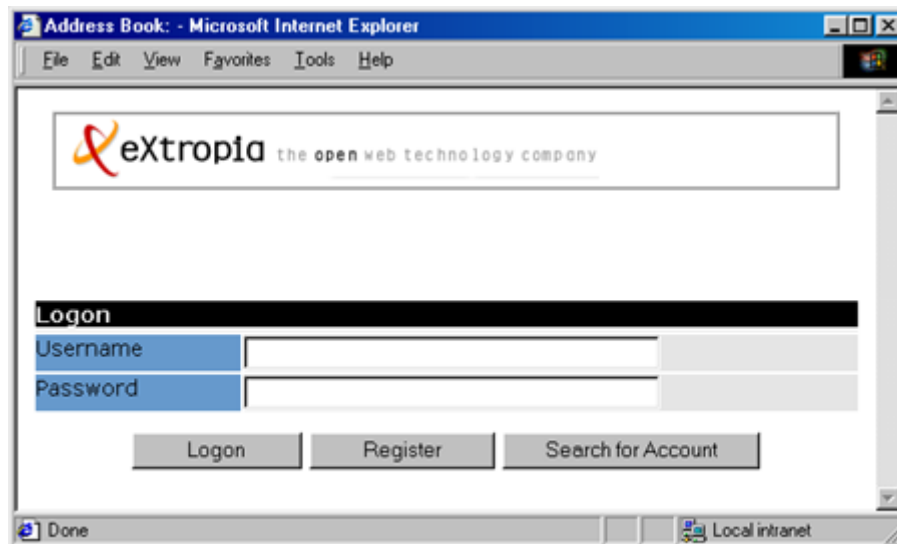
- Any user may view the database
- Any user may search the database
- Registered users may add, modify, and delete records
- Registration is available to anyone

Registration of normal users is simply done by pressing "Logon" and following the Registration Views. User data is written to the datasource. In the default installation, it is written to `Datafiles/InventoryManager/inventory_manager.users.dat`. It is recommended that administrators do not alter the contents, especially the encrypted passwords. The information is listed as:

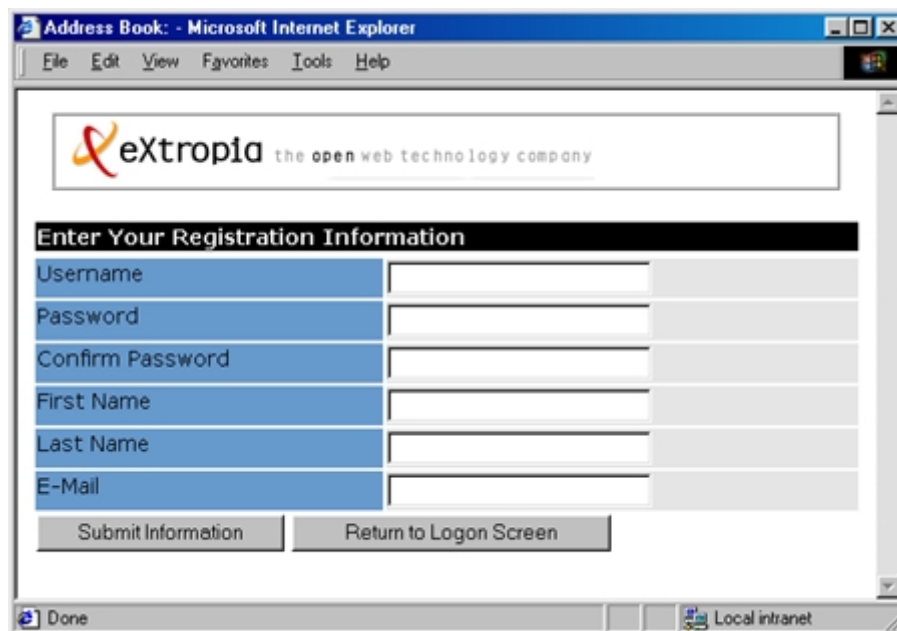
```
username | password | group_name | fname | lname | email
```

Registration of Administrative users may be done at a later date. System administrators will be able to alter hard data in databases or flat files as needed.

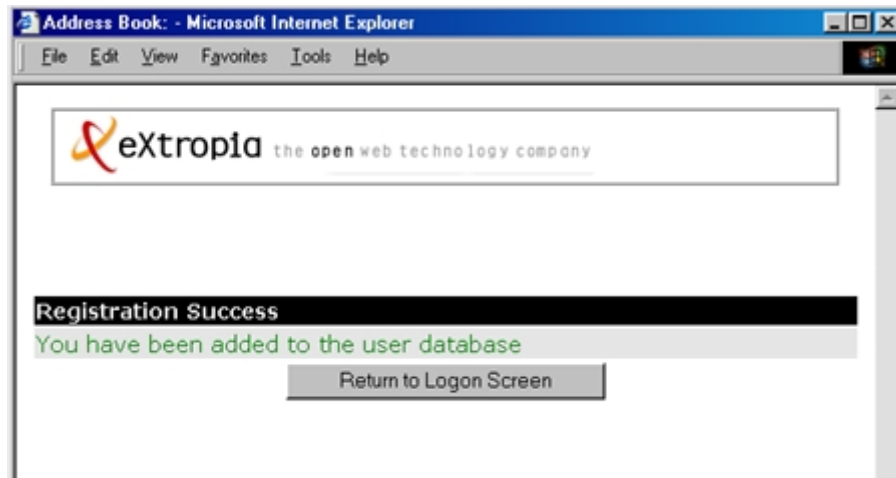
- **Logon View** requests the Username and Password of registered users. Users may attempt to logon, Register, or Search for Account. Also the view for failed logins.



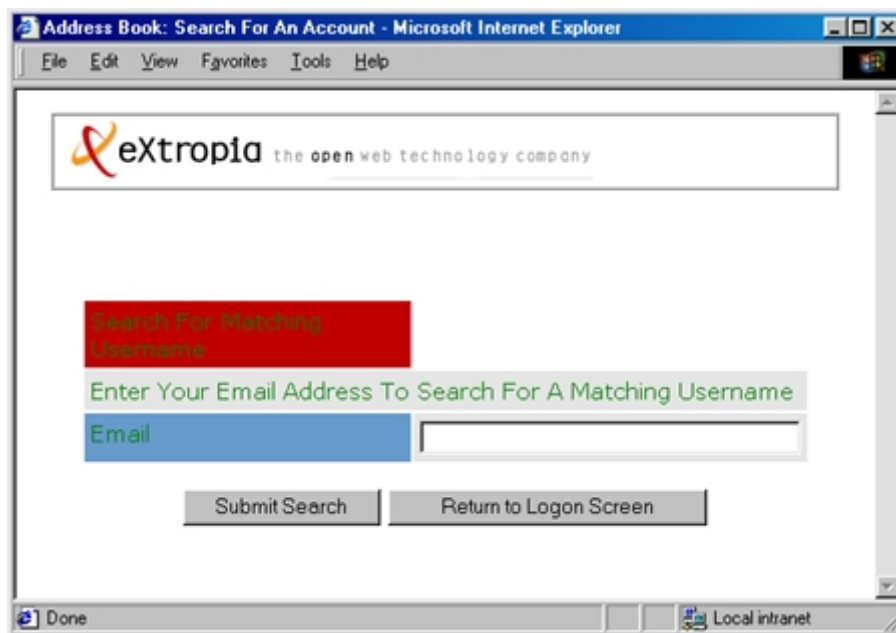
- **Register View** permits any user to enter Username, Password, Confirm, Password, First Name, Last Name, and E-Mail. The user may then Register or Return to Logon Screen.



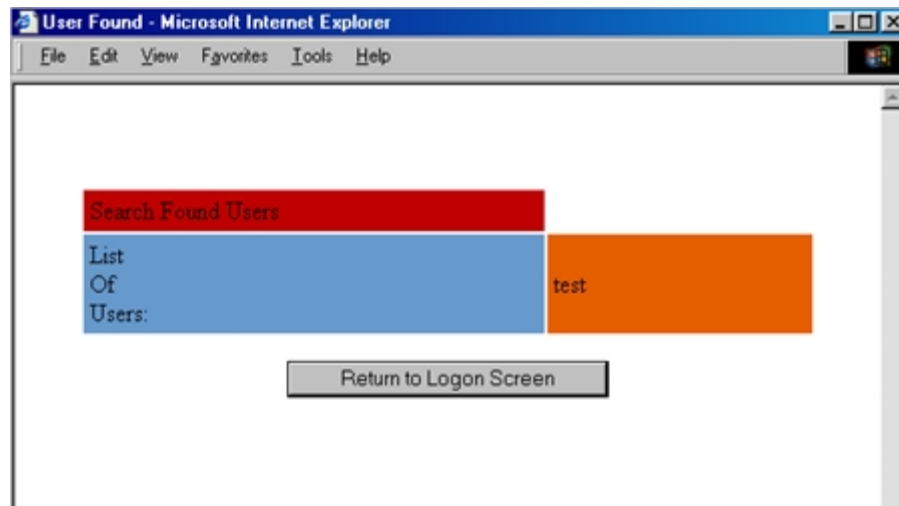
- **Register Confirmation View** confirms registration and returns the user to the Logon View.



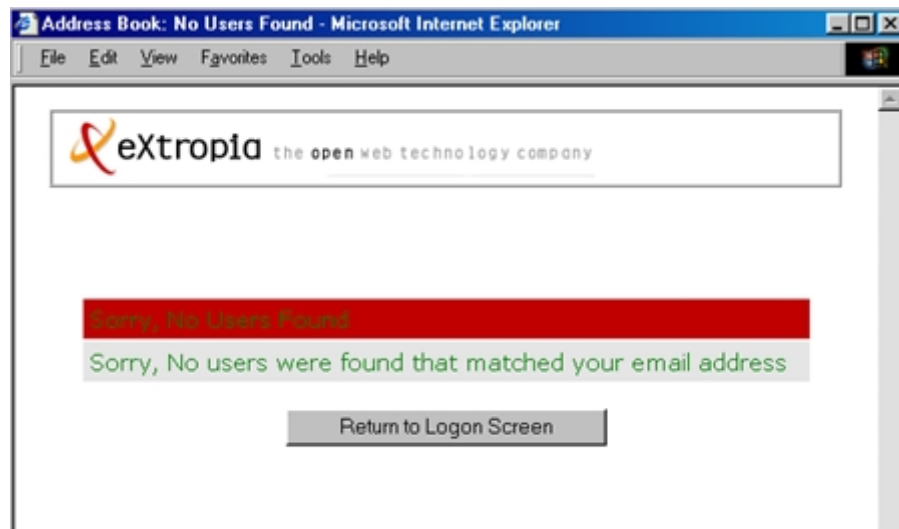
- **Search for Account** requests an email address to match users and then sends the password to the user.



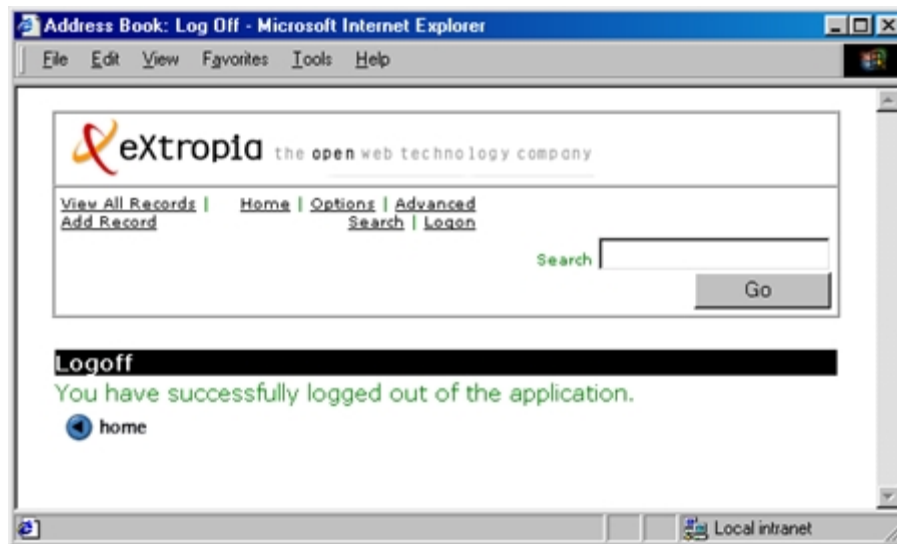
- **Search for Account Users Found** displays the username that matches the email address given.



- **Search for Users No Results** appears when no email address is associated with a username.

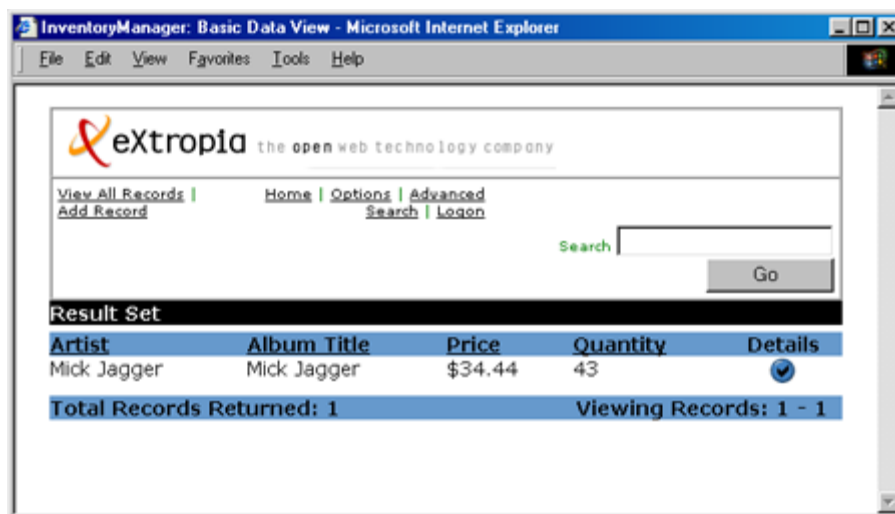


- **Logoff View** logs the user out and confirms the logoff. The session is terminated.



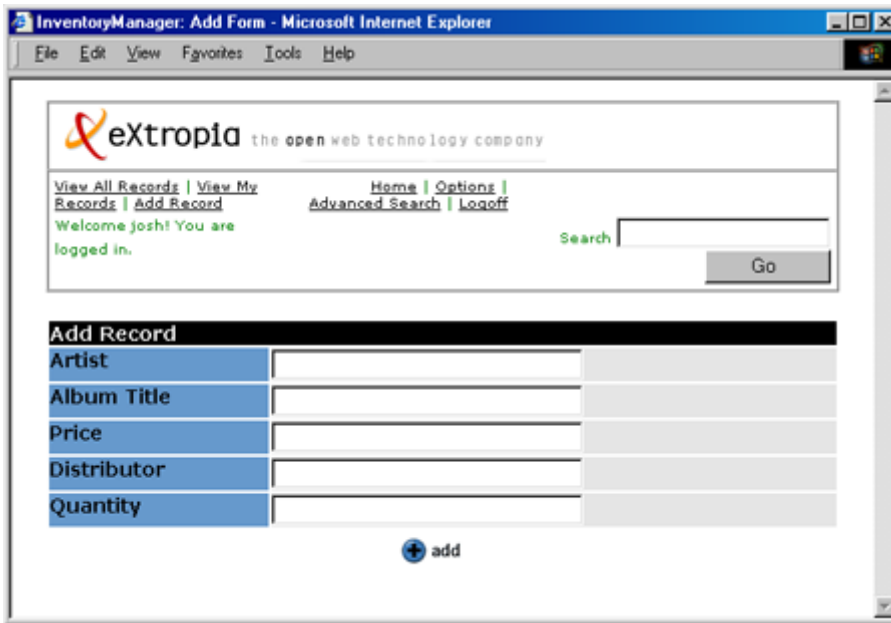
**Application Specific View Definitions**

- **Home View** will show all the information in the database ranked by First Name of the record in. This is the Default View. No authentication is required to View Records or Search the Database. The first 25 records are shown. Available options include Details, View All, View My Records, Add, Options, Advanced Search, Keyword Search, and Logoff.

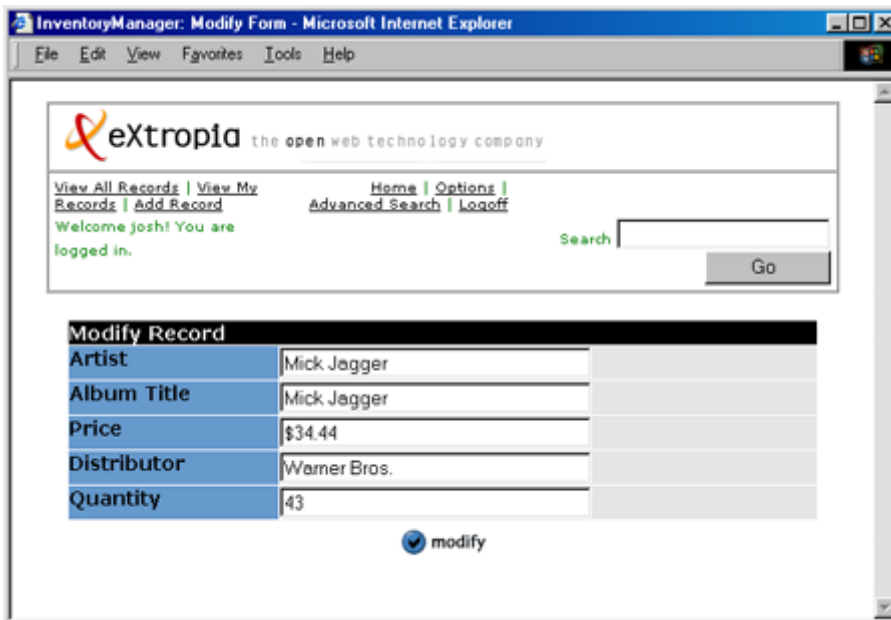


- **View All Records** will show all the information in the database ranked by First Name of the record in the address list.
- **View My Records** – allows a registered user to search within the subset of all records that were added by that user.

- **Add Record** requires login authentication to Add Records. Fields include: Artist, Album Title, Price, Distributor, Quantity.

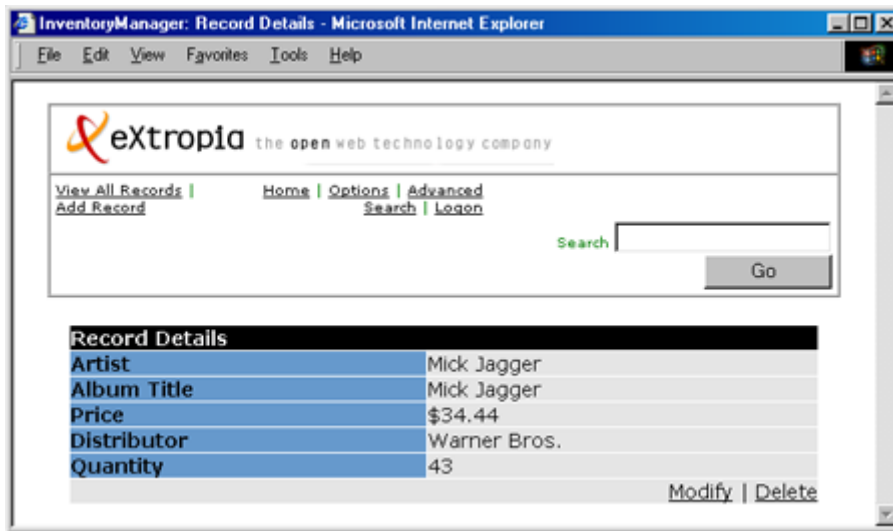


- **Delete Record** requires authentication. It will ask the user to confirm.
- **Modify Record** requires authentication. It contains the same fields as Add Record.

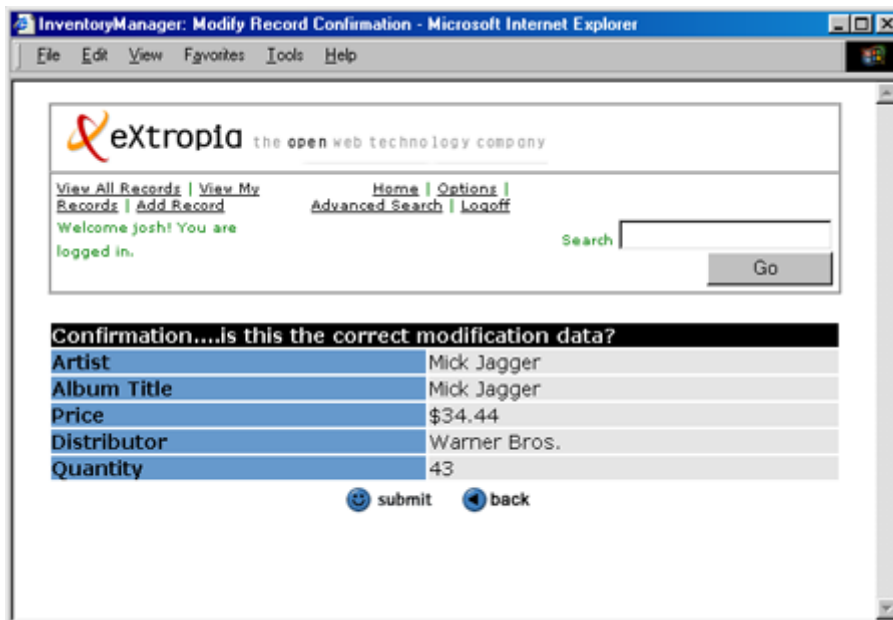




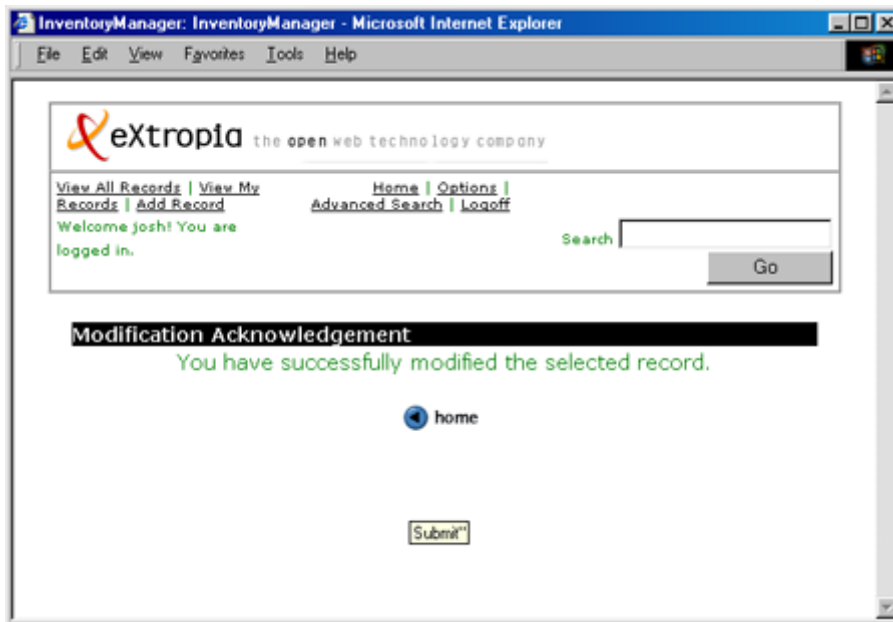
- **Details View** displays the entire contents of a record. This view permits Modify and Delete with authentication.



- **Confirmation View** is displayed whenever someone attempts to Add, Modify, or Delete a Record.

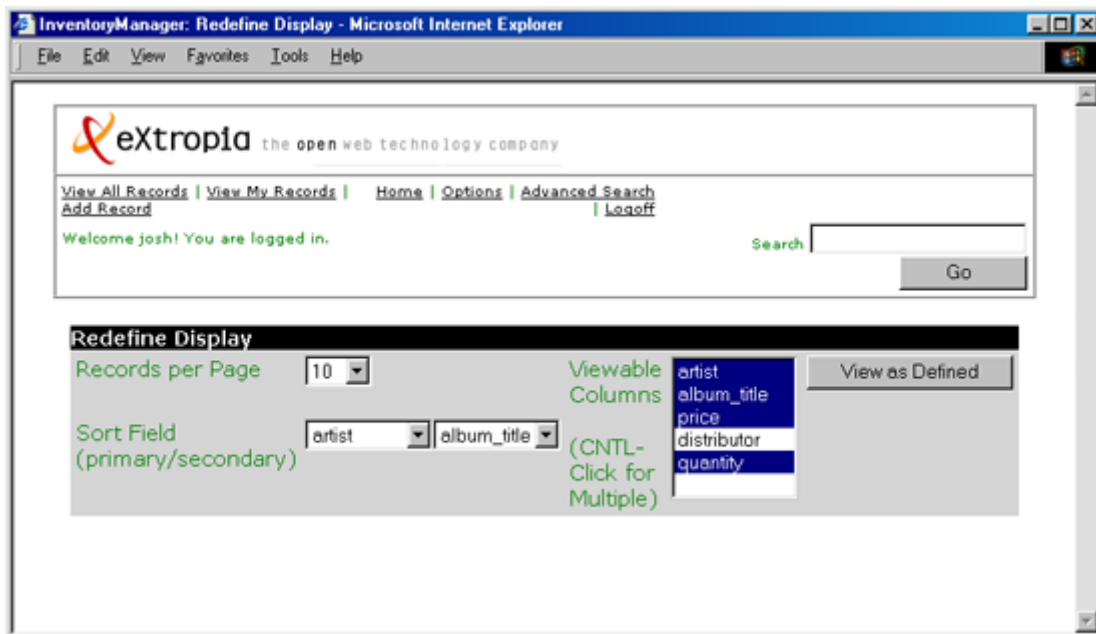


- **Acknowledgment View** is displayed whenever someone successfully Adds, Modifies, or Deletes a Record. A link to the Home View is available.



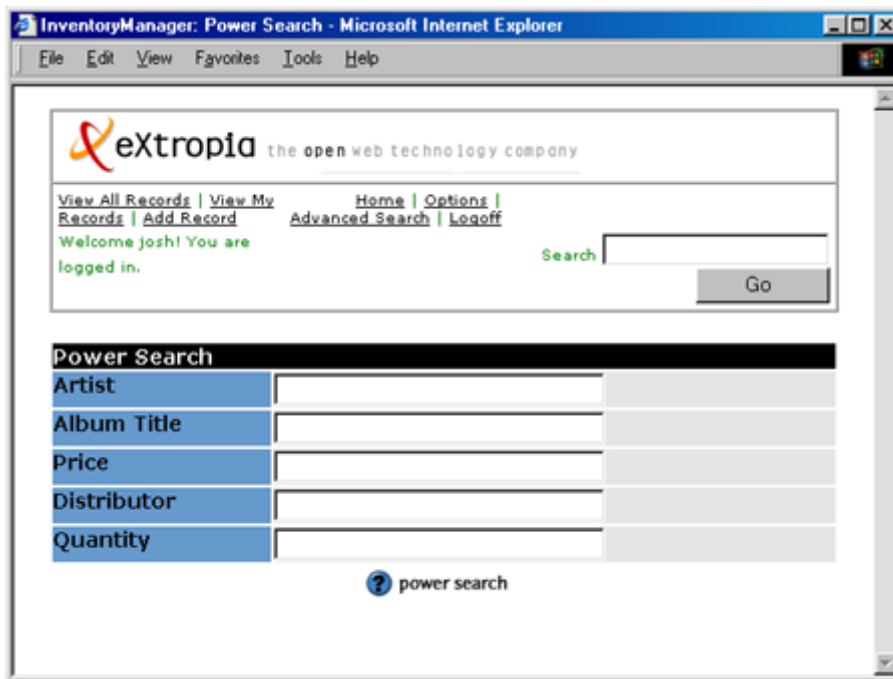
**Common Features Definitions**

- **Options View** permits the authenticated user to alter Records Displayed per Page, Two Sort Fields, and which Columns to View. Upon pressing View as Defined, the user will see the records as desired. The Viewable Columns will vary according to Application.



- **Basic Search** permits any user to search the database based upon keywords. Available in the header.

- **Advanced Search** permits any user to search the database for any or all of the available fields



### 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 )
artist	Store the artist of the album record.
album_title	Store the album title of the record.
price	Store the price of each record.
distributor	Store the distributor name who distribute this record.
quantity	Store the quantity available in the inventory list.
username_of_poster	Store the user_id of the user who upload the file record.
group_of_poster	Store the group of the login user who upload the record. Eg ( normal user, admin user. )
date_time_posted	Store the date and time at the posting of this record

### 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 flat-file 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.

A modular architecture provides easy maintenance and upgrades.

### 5.3 Portability

### 5.4 Extensibility

Information entered via Web Inventory Manager 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

Web Inventory Manager is highly reusable. The application can be modified for just about anything!

### 5.6 Serviceability