US 20110071996A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2011/0071996 A1

(43) Pub. Date: Mar. 24, 2011

Fisher et al.

(54) **SYSTEMS AND METHODS FOR IMPLEMENTING MULTI-APPLICATION TABS AND TAB SETS**

(75) Inventors: **Steve Fisher**, San Francisco, CA (US); **Eric David Bezar**, Oakland, CA (US); **Adam Gross**, San Francisco, CA (US); **Ian Swinson**, Oakland, CA (US); **George Hu**, San Francisco, CA (US); **Yoshiyuki Oikawa**, San Francisco, CA (US)

(73) Assignee: **salesforce.com, inc.**, San Francisco, CA (US)

(21) Appl. No.: **12/957,258**

(22) Filed: **Nov. 30, 2010**

**Related U.S. Application Data**

(63) Continuation of application No. 12/831,072, filed on Jul. 6, 2010, which is a continuation of application No. 11/075,546, filed on Mar. 8, 2005, now Pat. No. 7,774,366.

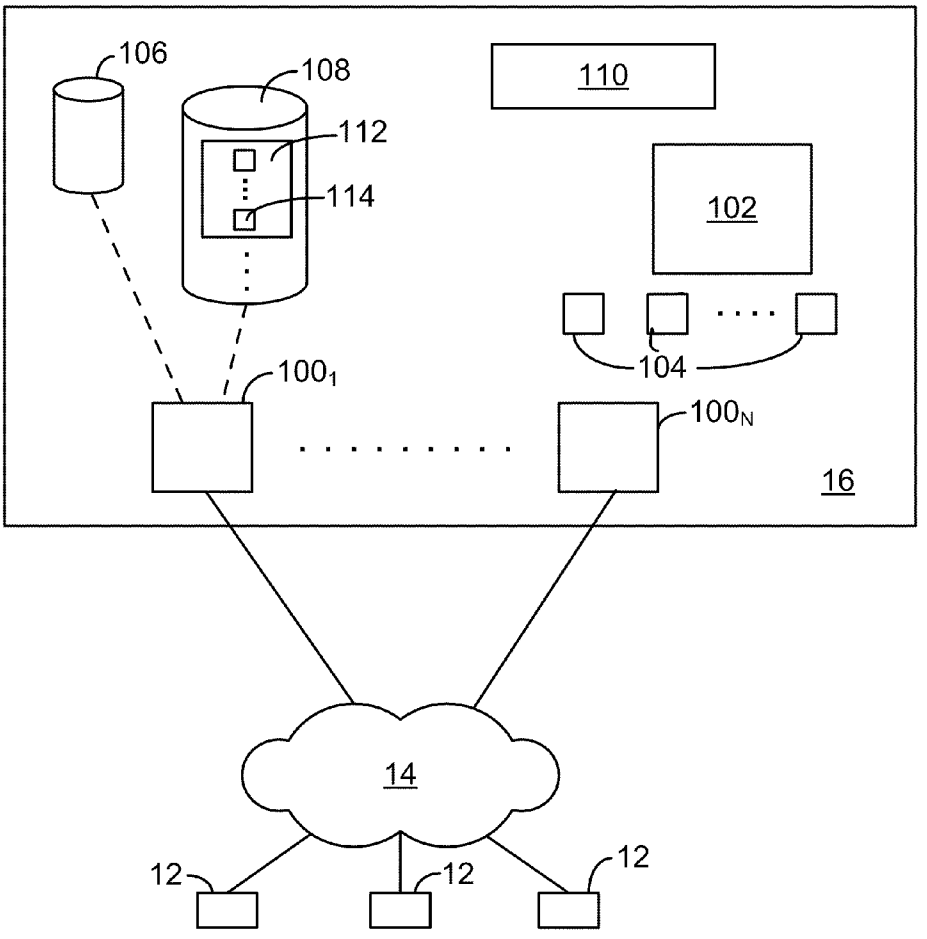
**Publication Classification**

(51) **Int. Cl.**
*G06F 3/048* (2006.01)
*G06F 17/30* (2006.01)

(52) **U.S. Cl.** ..... **707/705**; 715/777; 707/E17.005; 707/E17.044

(57) **ABSTRACT**

A sharing model enables users to easily and conveniently create and share standard and custom database objects, and applications, among users or groups of users. The sharing model includes systems and methods that allow users to conveniently create and customize tabs and tab sets that define relationships between custom objects and fields, standard objects and fields, and applications. A tab visually represents a user interface into an element of an application or into a database object. Selection of a tab provides a user access to the object or element of the application represented by the tab. A tab set is a group of related tabs that work as a unit to provide application functionality. New tabs and tab sets may be defined and tab set views may be customized so that an end user can easily and conveniently switch between the various objects and application elements represented by the displayed tabs and tab set views.



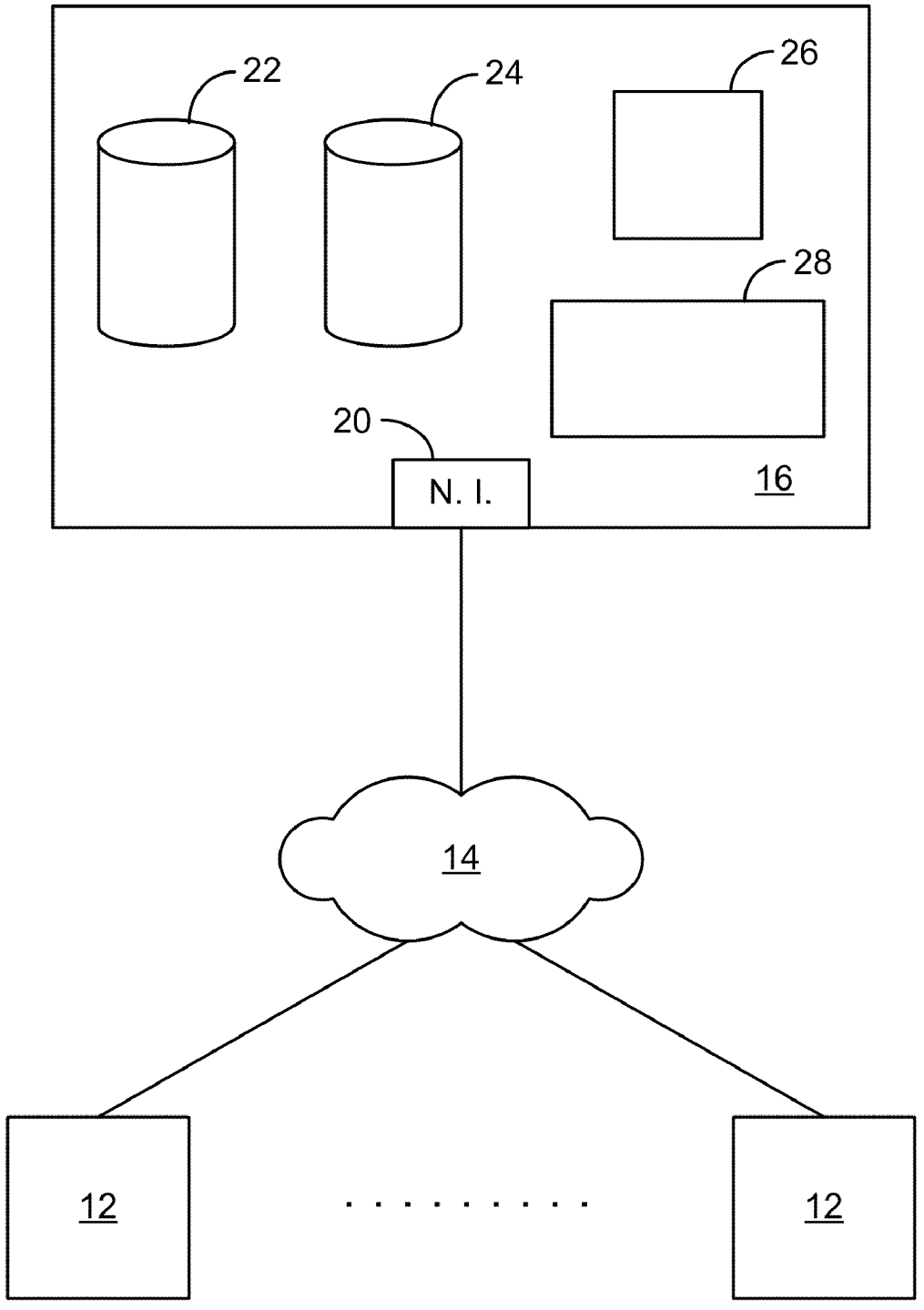



*FIG. 1*

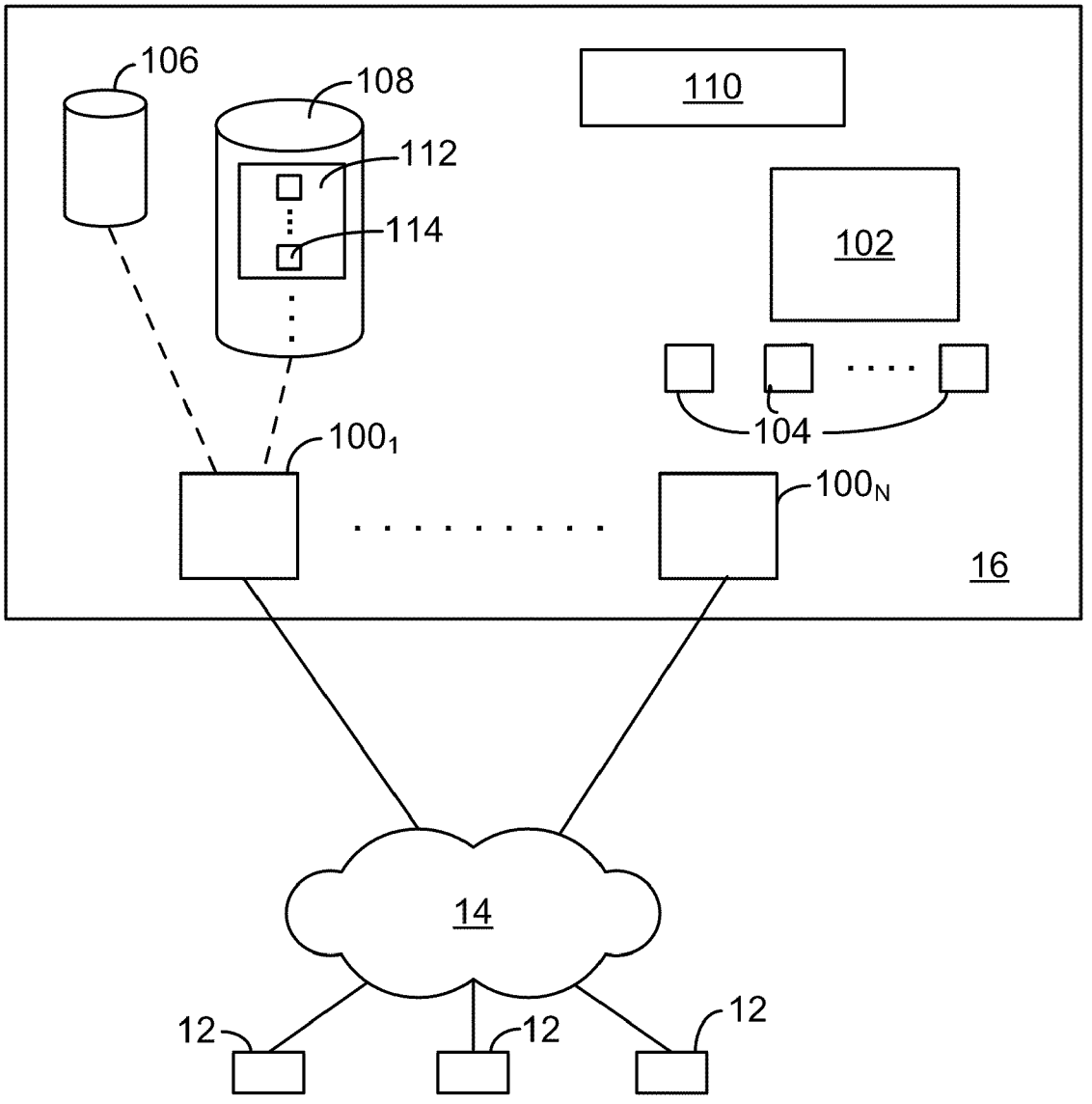



*FIG. 2*

FIG. 3

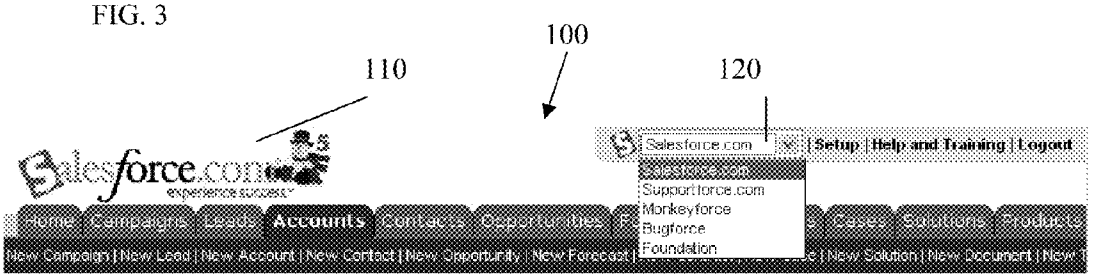

100
110
120
Salesforce.com
Setup | Help and Training | Logout
Salesforce.com
Supportforce.com
Monkeyforce
Bugforce
Foundation
Home
Campaigns
Leads
Accounts
Contacts
Opportunities
Cases
Solutions
Products
New Campaign | New Lead | New Account | New Contact | New Opportunity | New Forecast | New Solution | New Document

FIG. 4: Tab Set Logos and Selector

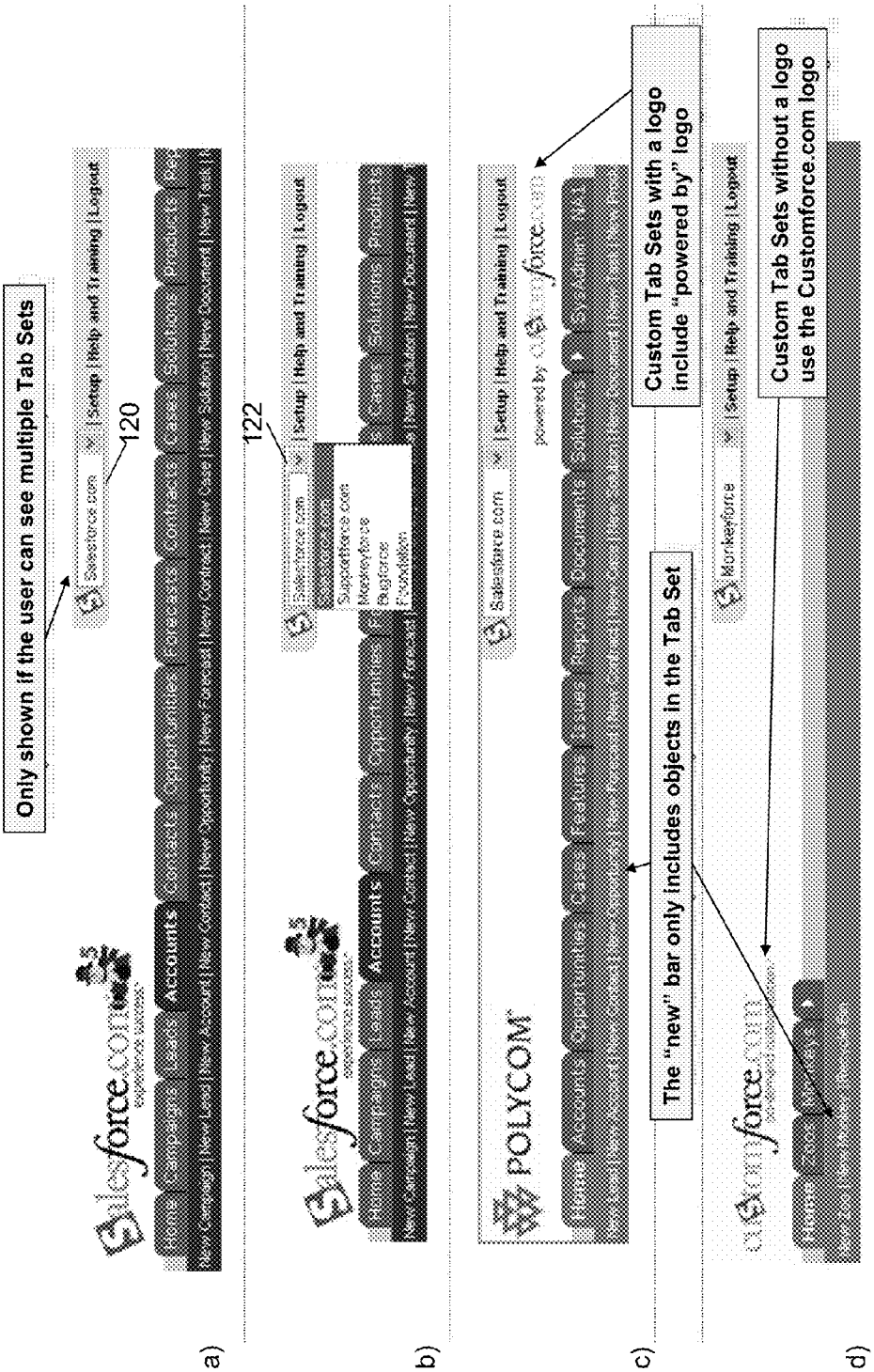


FIG. 5: All Tabs Page

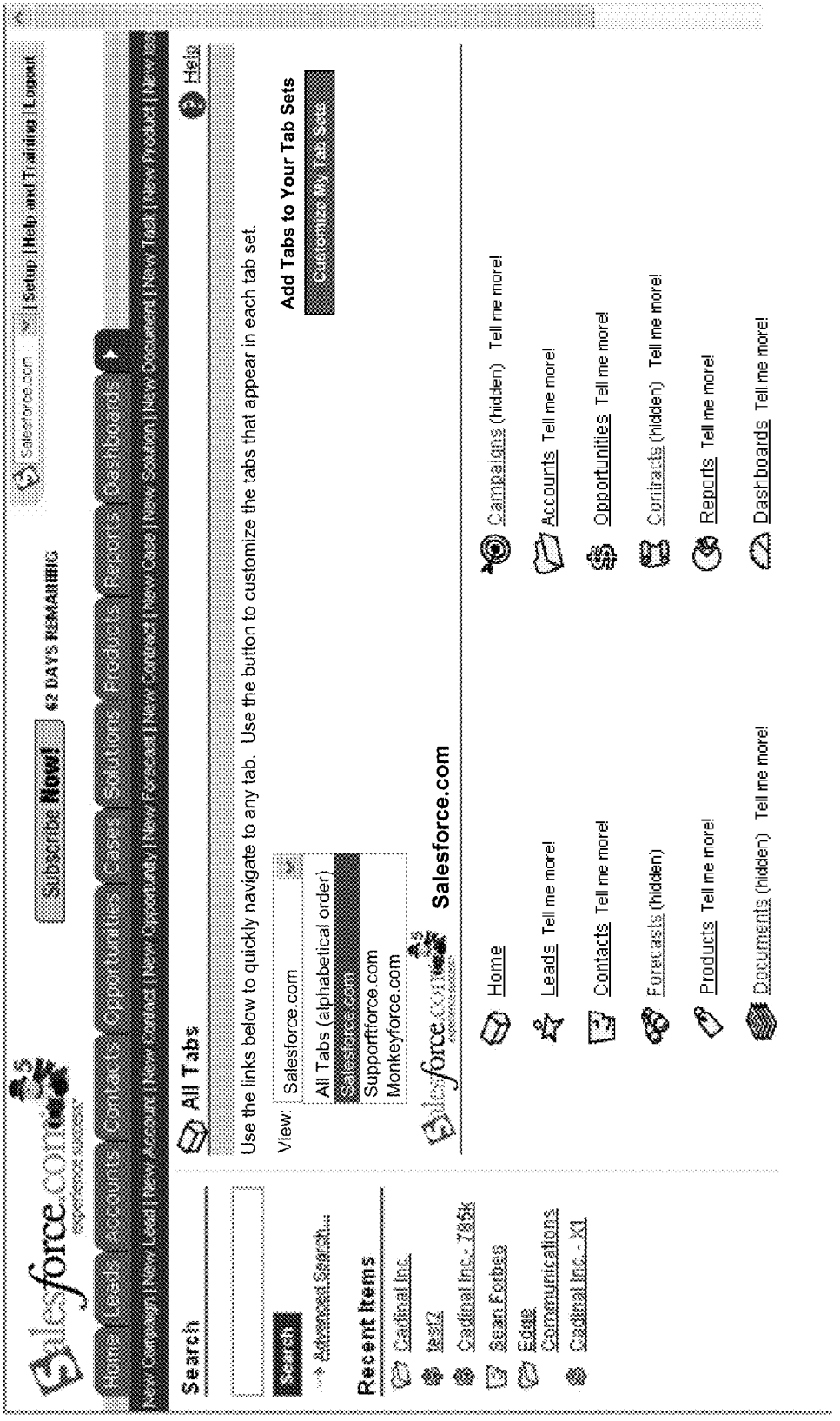


FIG. 6: Customize My Tabs

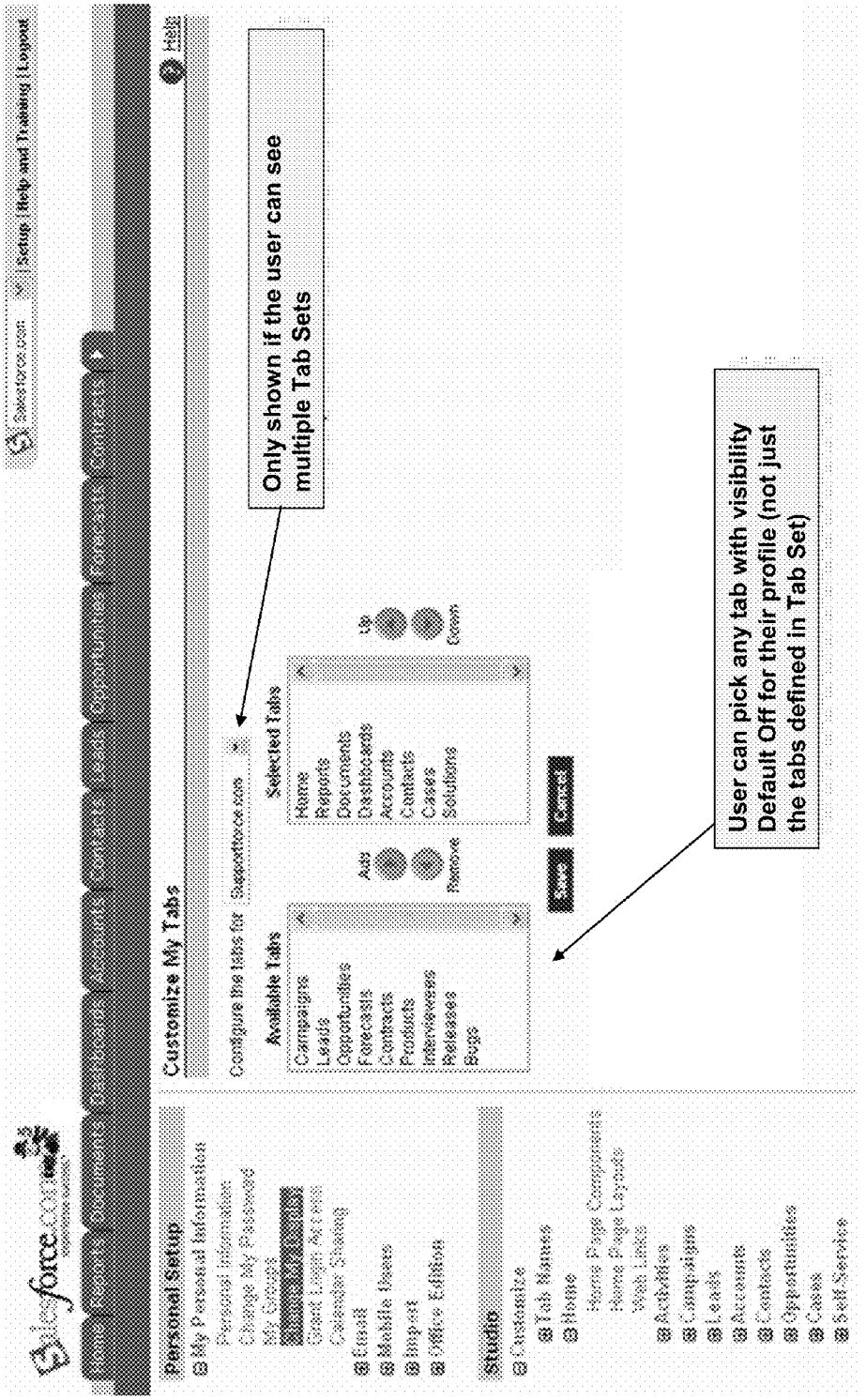


FIG. 7: Setup → Customize → Tabs → Tab Sets

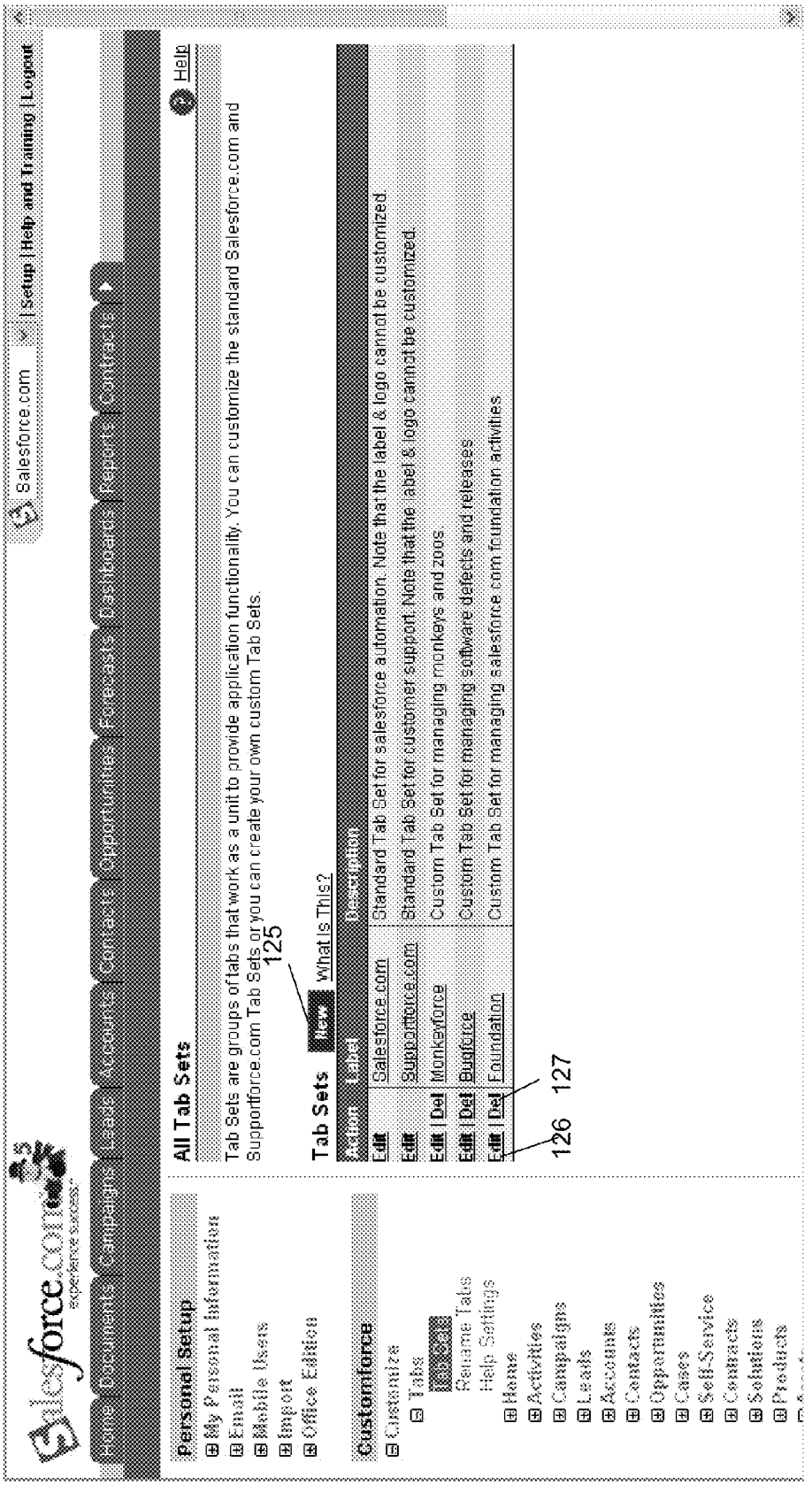


FIG. 8: Tab Set Wizard – Step 1 (Custom Tab Sets)

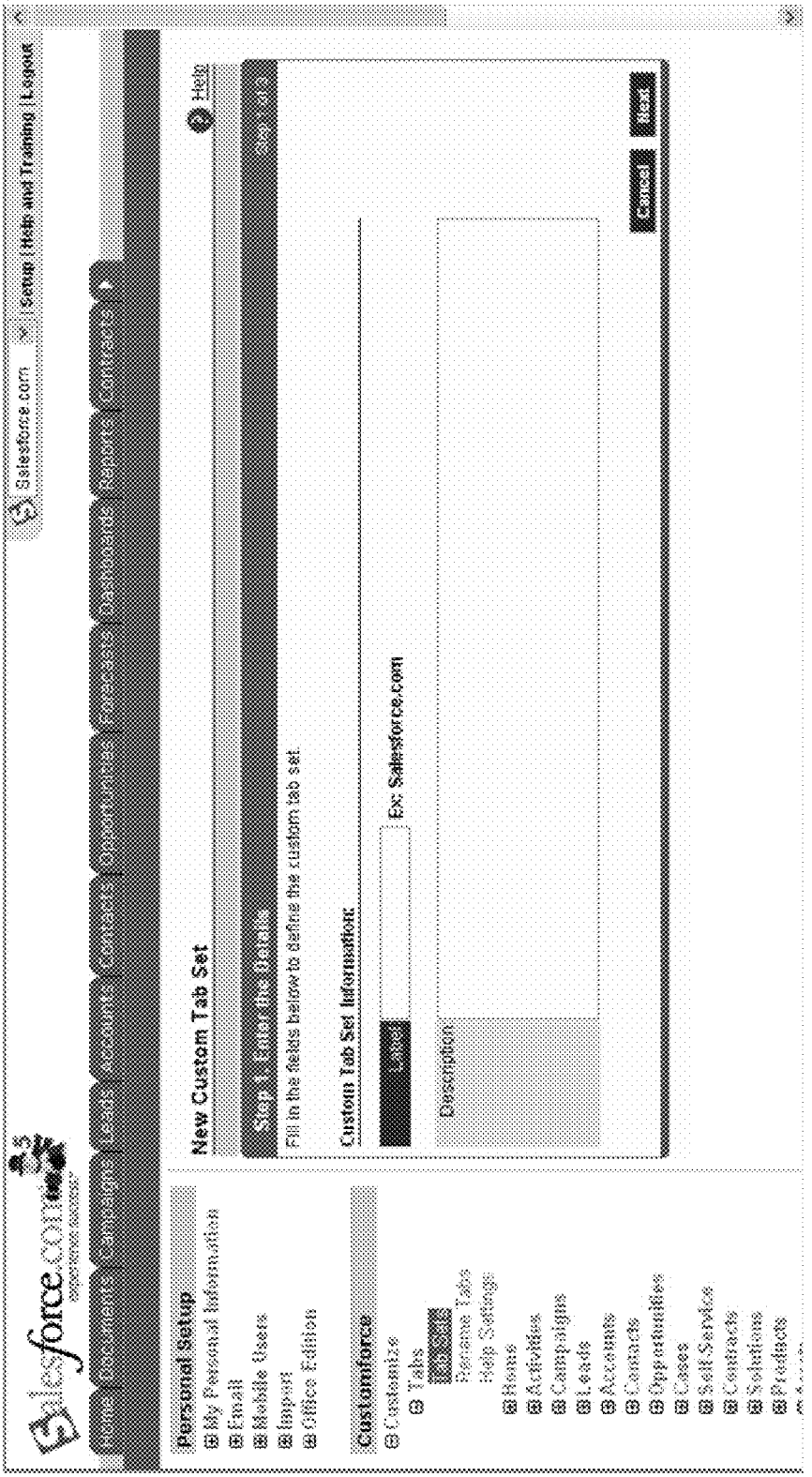


FIG. 9: Tab Set Wizard – Step 2 (Custom Tab Sets)

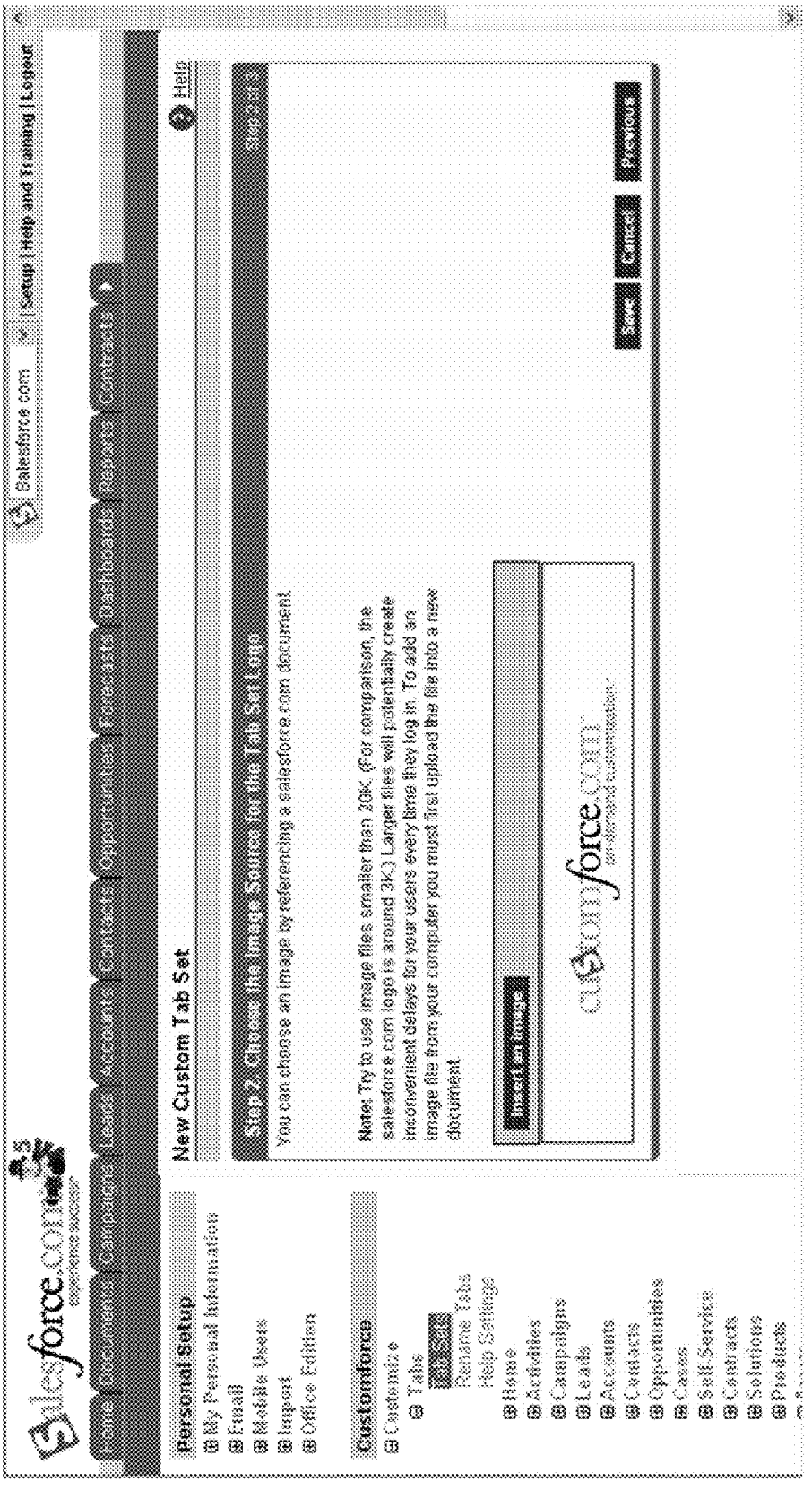


FIG. 10: Tab Set Wizard – Step 3 (Custom Tab Sets)

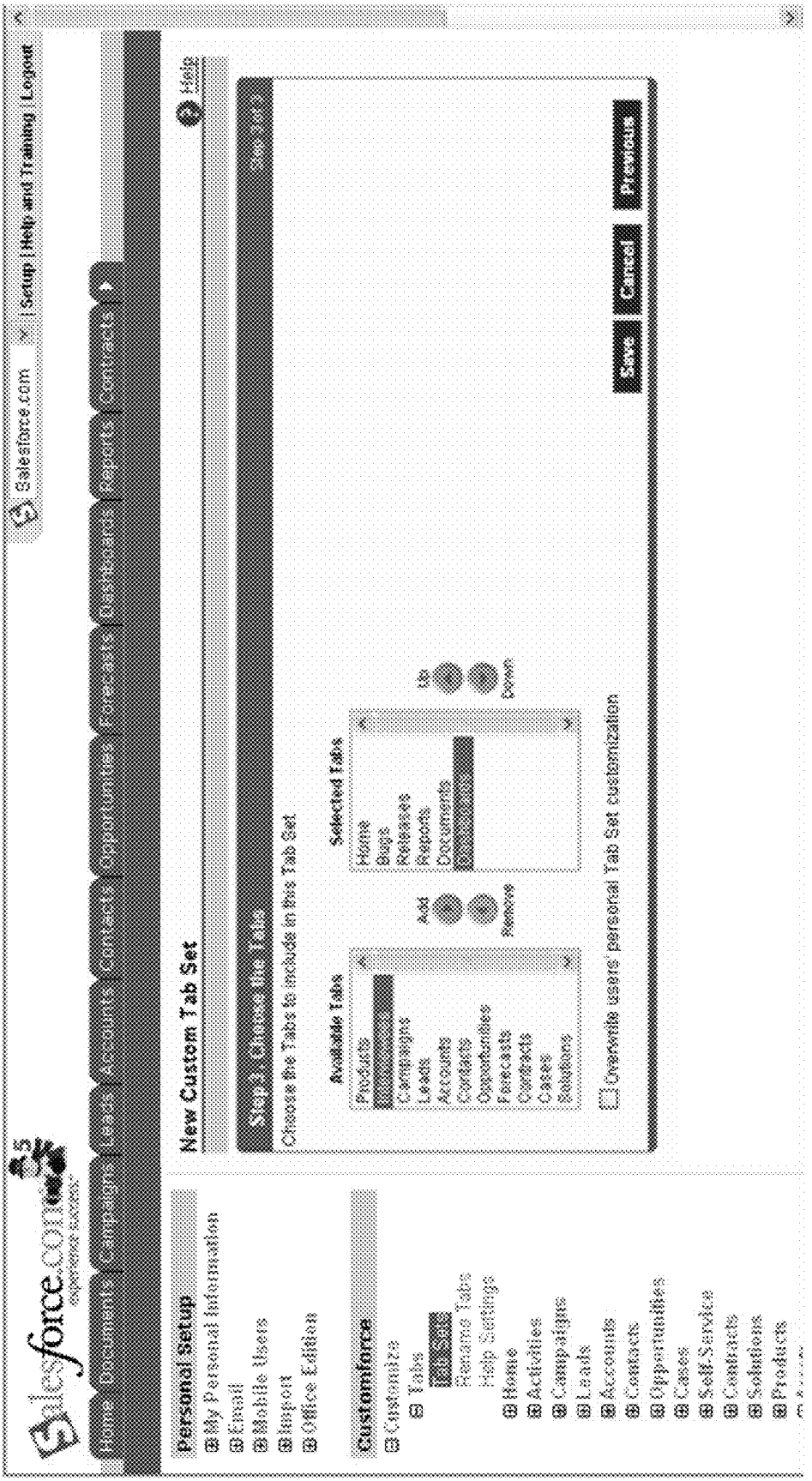


**FIG. 11: Tab Set Wizard – Step 4 (Custom Tab Sets)**

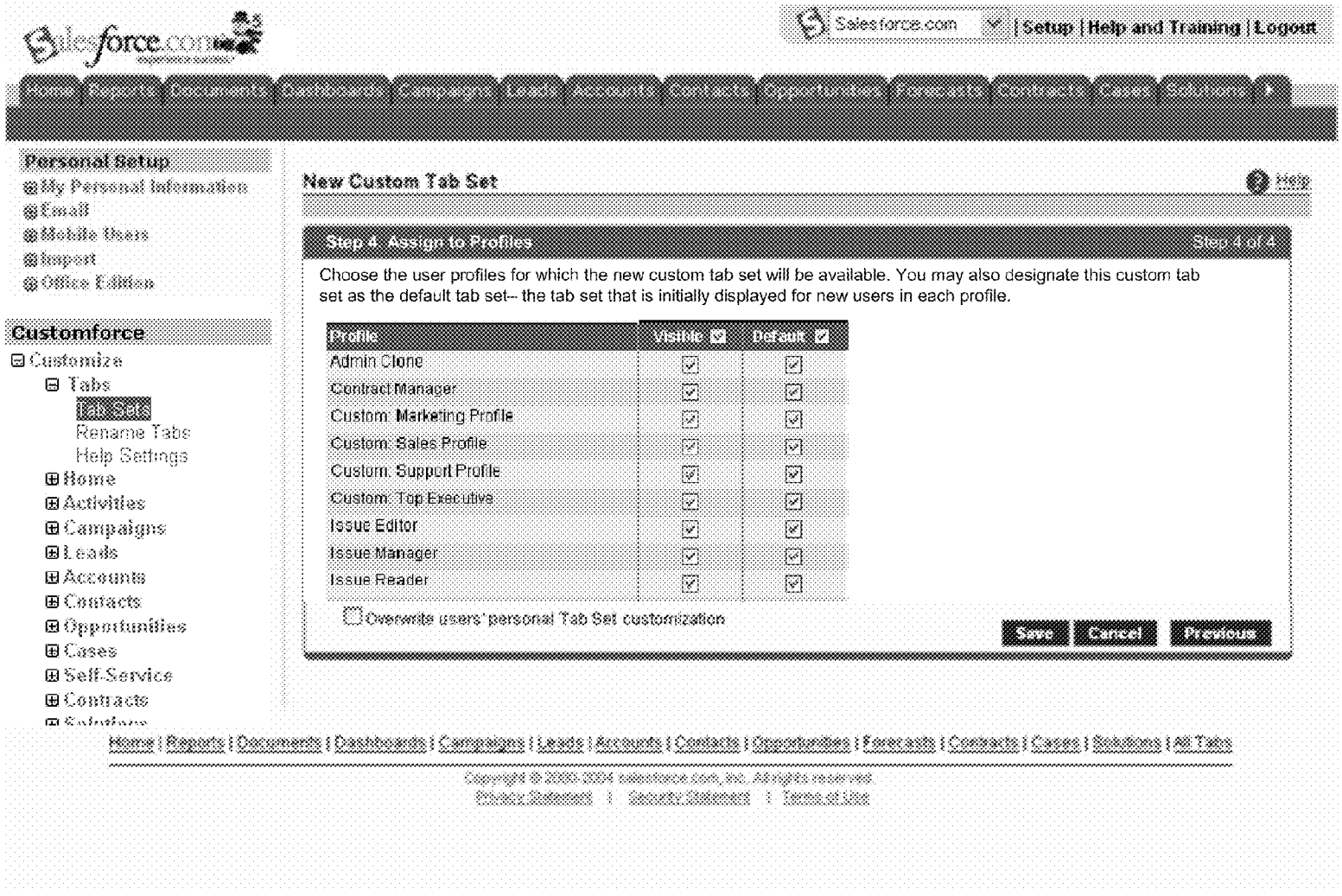


FIG. 12: Tab Set Detail Page

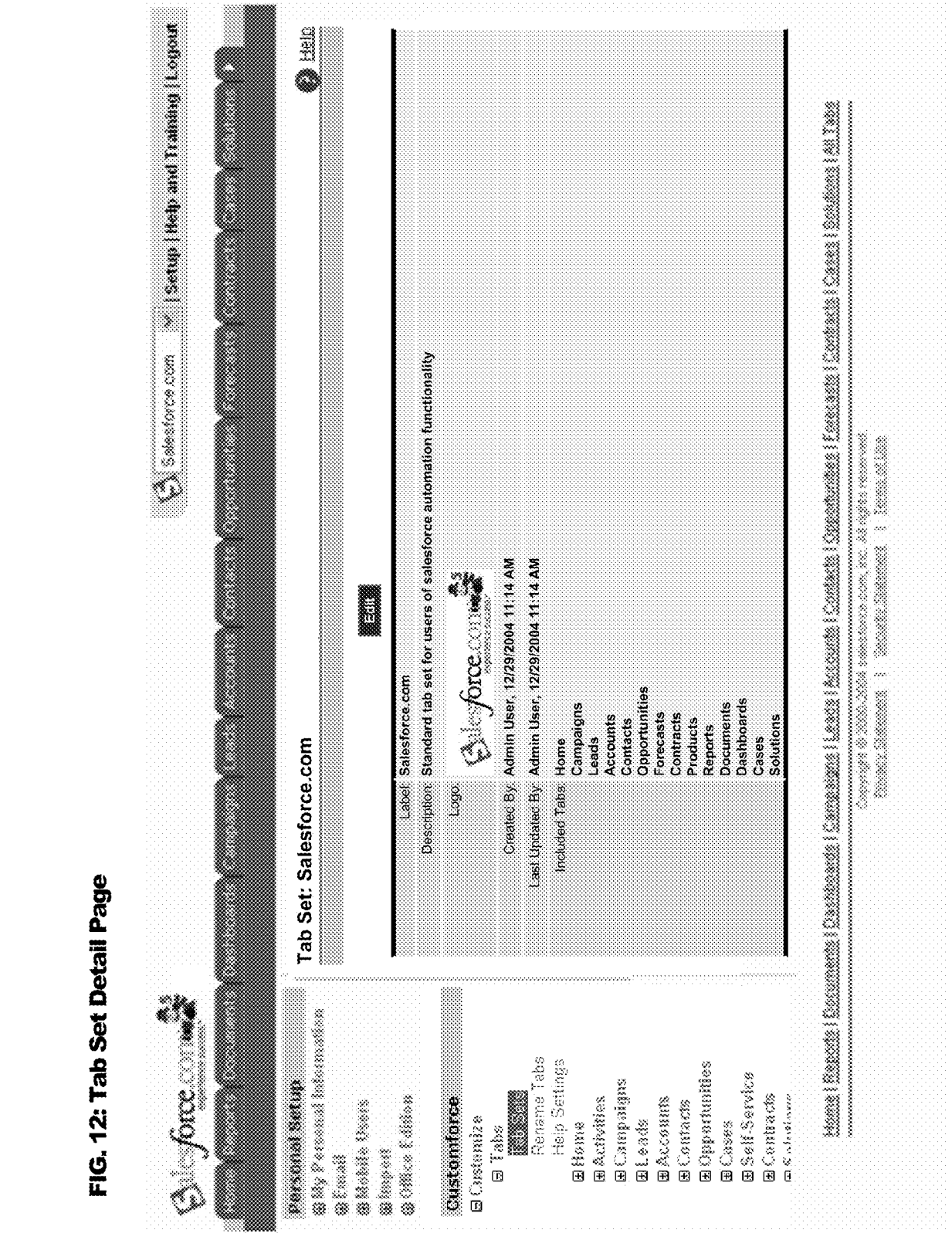


**FIG. 13: Delete Tab Set – Replace Deleted Value**

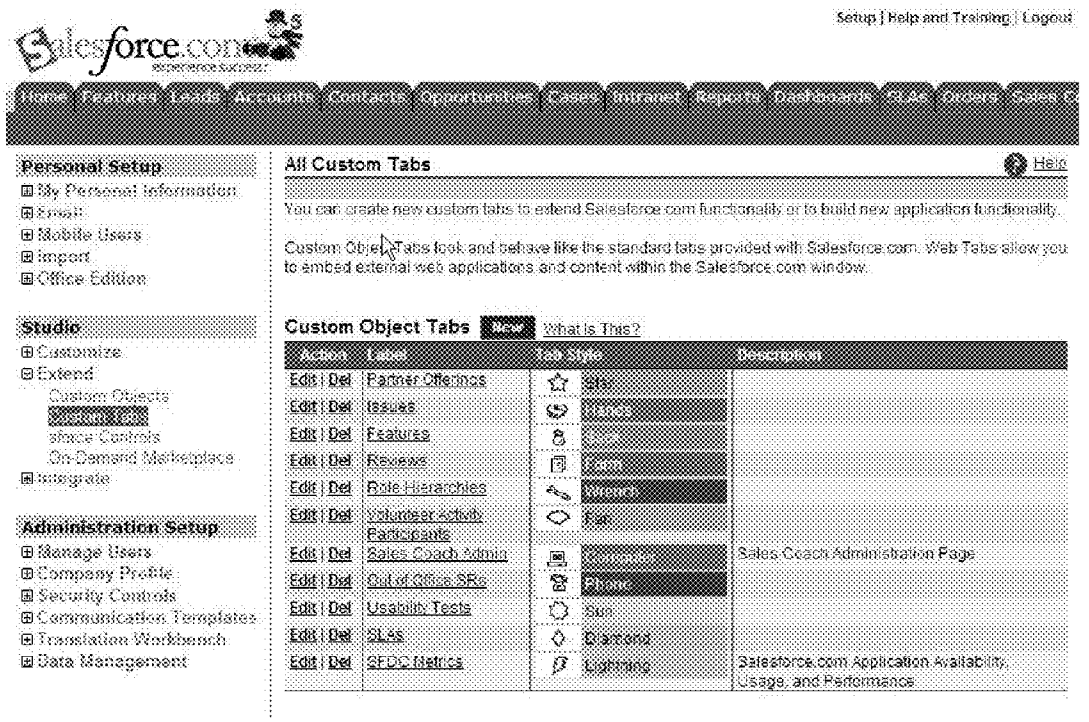

Setup | Help and Training | Logout
Home | Leads | Accounts | Contacts | Opportunities | Cases | Reports | Dashboards
Personal Setup
My Personal Information
Email
Mobile Users
Import
Office Edition
Studio
Customize
Extend
Custom Objects
Custom Tabs
Integrate
Administration Setup
Manage Users
Company Profile
Security Controls
Communication Templates
Translation Workbench
Data Management
All Custom Tabs
Help
You can create new custom tabs to extend Salesforce.com functionality or to build new application functionality.
Custom Object Tabs look and behave like the standard tabs provided with Salesforce.com. Web Tabs allow you to embed external web applications and content within the Salesforce.com window.
Custom Object Tabs
New
What is This?
Action
Label
Tab Style
Description
Edit | Del
Partner Offerings
Issues
Features
Reviews
Role Hierarchies
Volunteer Activity Participants
Sales Coach Admin
Sales Coach Administration Page
Out of Office SRs
Usability Tests
SLAs
SFDC Metrics
Salesforce.com Application Availability, Usage, and Performance

FIG. 14

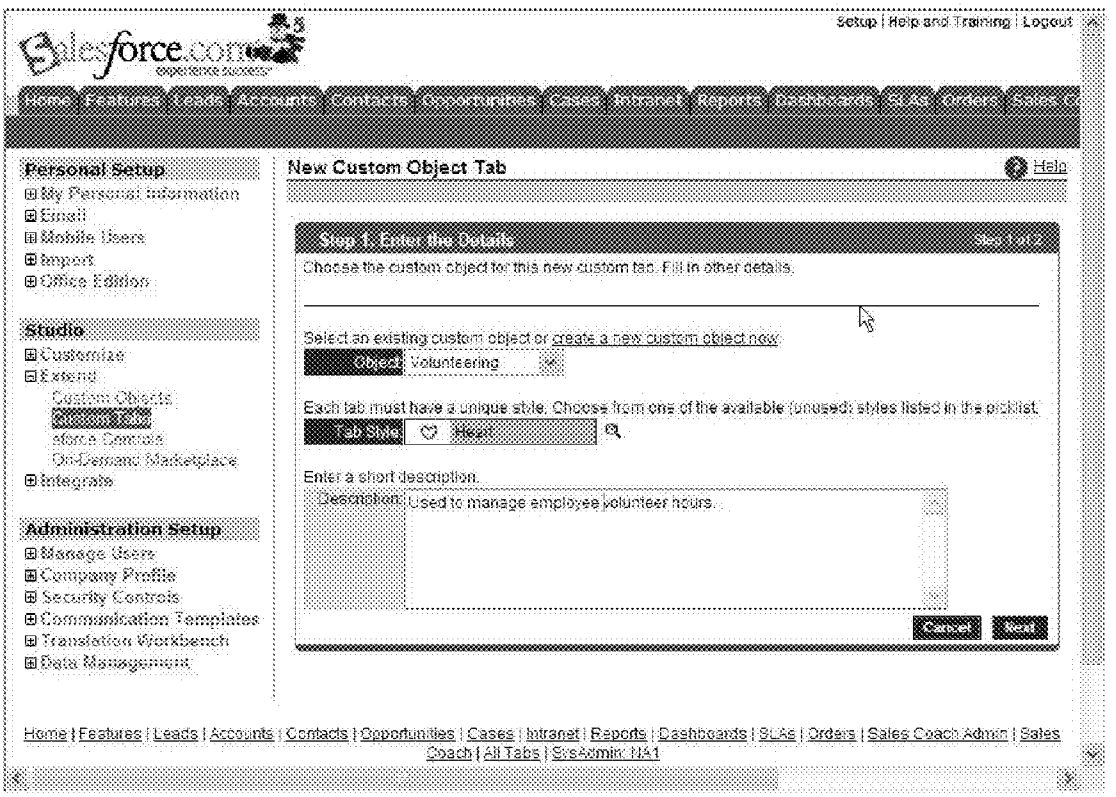

Setup | Help and Training | Logout
Home | Features | Leads | Accounts | Contacts | Opportunities | Cases | Intranet | Reports | Dashboards | SLAs | Orders | Sales Co
Personal Setup
My Personal Information
Email
Mobile Users
Import
Office Edition
Studio
Customize
Extend
Custom Objects
Custom Tabs
Force Controls
On-Demand Marketplace
Integrate
Administration Setup
Manage Users
Company Profile
Security Controls
Communication Templates
Translation Workbench
Data Management
New Custom Object Tab
Help
Step 1. Enter the Details
Step 1 of 2
Choose the custom object for this new custom tab. Fill in other details.
Select an existing custom object or create a new custom object now
Object: Volunteering
Each tab must have a unique style. Choose from one of the available (unused) styles listed in the picklist.
Tab Style:
Enter a short description.
Description: Used to manage employee volunteer hours.
Cancel
Next
Home | Features | Leads | Accounts | Contacts | Opportunities | Cases | Intranet | Reports | Dashboards | SLAs | Orders | Sales Coach Admin | Sales Coach | All Tabs | SysAdmin: NA1

FIG. 15

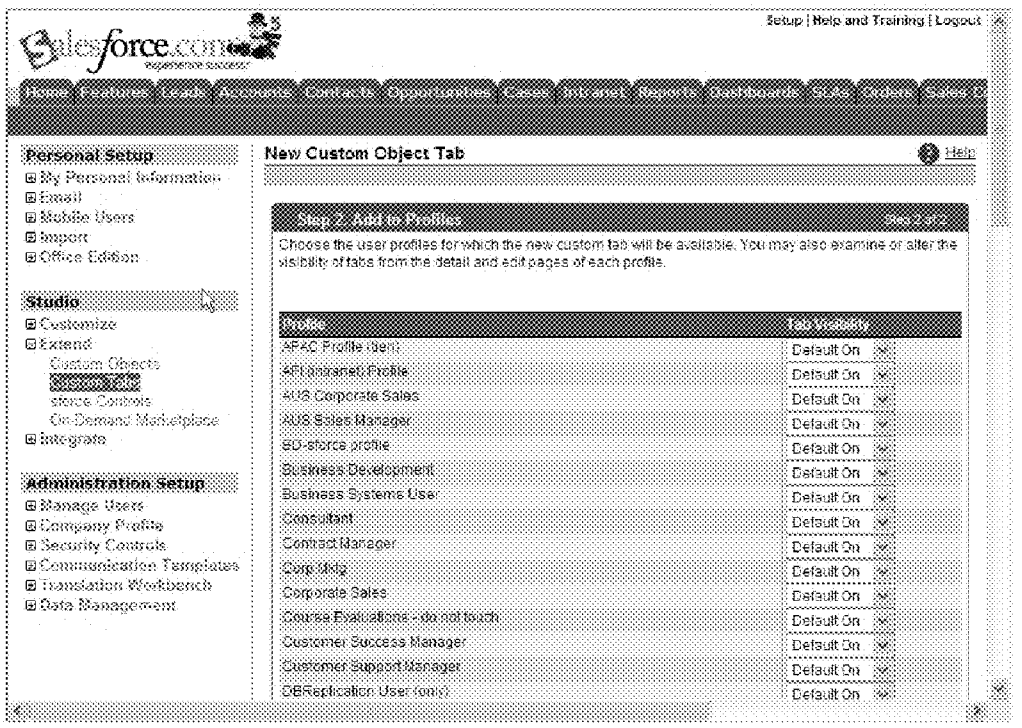

Setup | Help and Training | Logout
Personal Setup
My Personal Information
Email
Mobile Users
Import
Office Edition
Studio
Customize
Extend
Custom Objects
Integrate
Administration Setup
Manage Users
Company Profile
Security Controls
Communication Templates
Translation Workbench
Data Management
New Custom Object Tab
Help
Choose the user profiles for which the new custom tab will be available. You may also examine or alter the visibility of tabs from the detail and edit pages of each profile.
Profile
Tab Visibility
APAC Profile (test)
API Intranet Profile
AUS Corporate Sales
AUS Sales Manager
BD-sforce profile
Business Development
Business Systems User
Consultant
Contract Manager
Corp Mktg
Corporate Sales
Course Evaluations - do not touch
Customer Success Manager
Customer Support Manager
Default On

FIG. 16

FIG. 17: Custom Tab Wizard – Step 3 – Add to Tab Sets

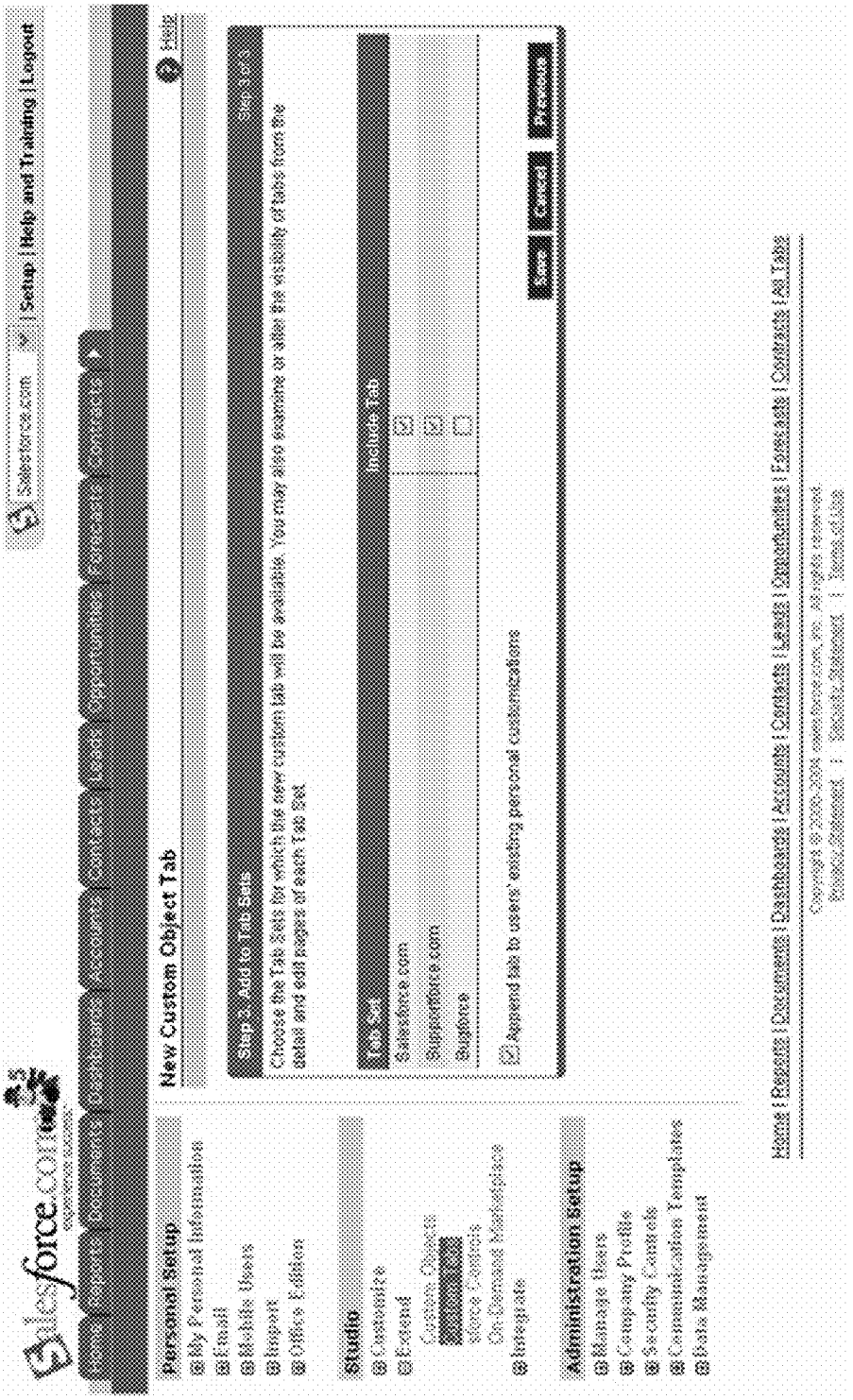
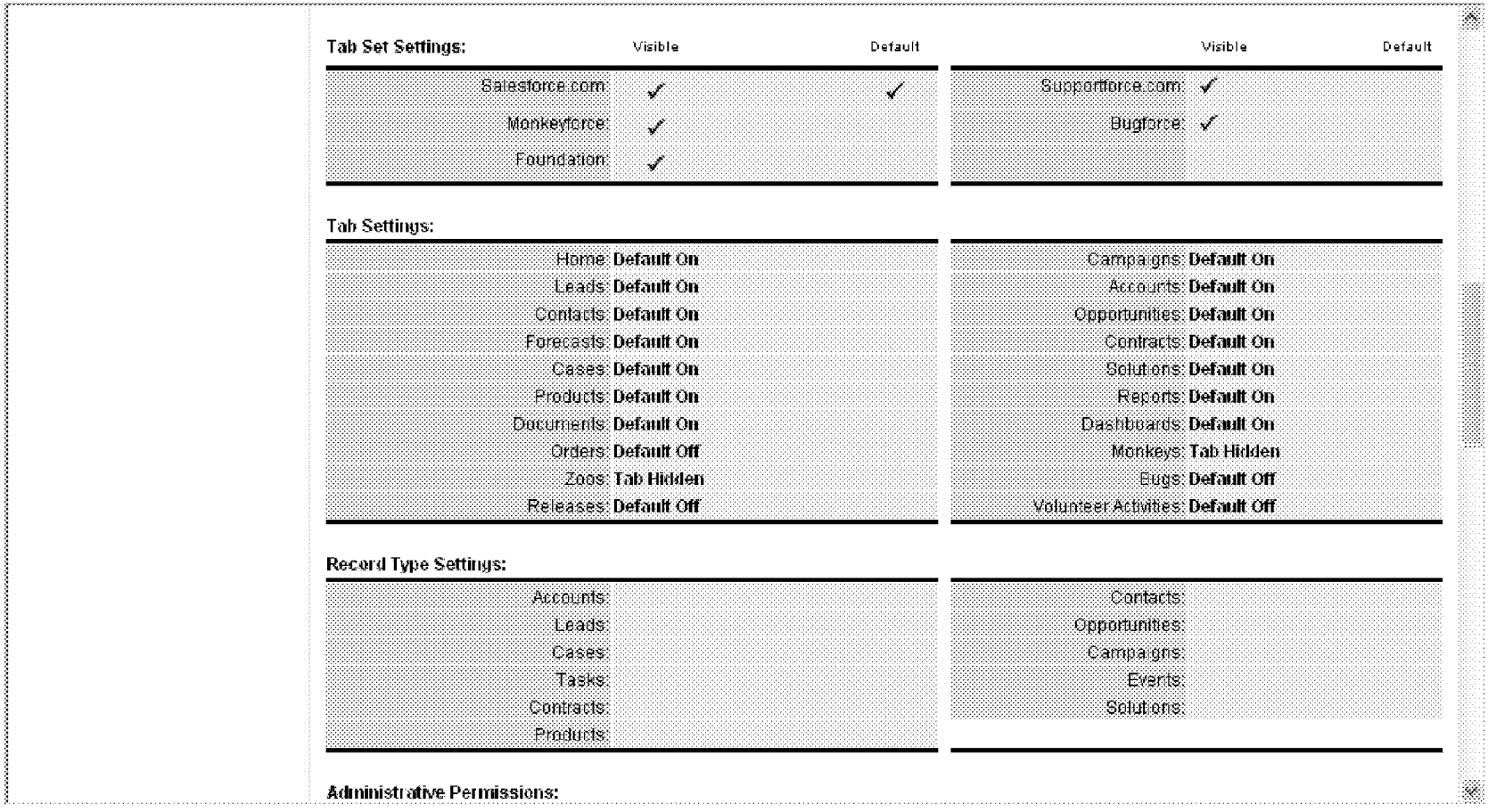


**FIG. 18: Profile Detail Page**



**Tab Set Settings:**

| | Visible | Default | | Visible | Default |
|---|---|---|---|---|---|
| Salesforce.com: | ✓ | ✓ | Supportforce.com: | ✓ | |
| Monkeyforce: | ✓ | | Bugforce: | ✓ | |
| Foundation: | ✓ | | | | |

**Tab Settings:**

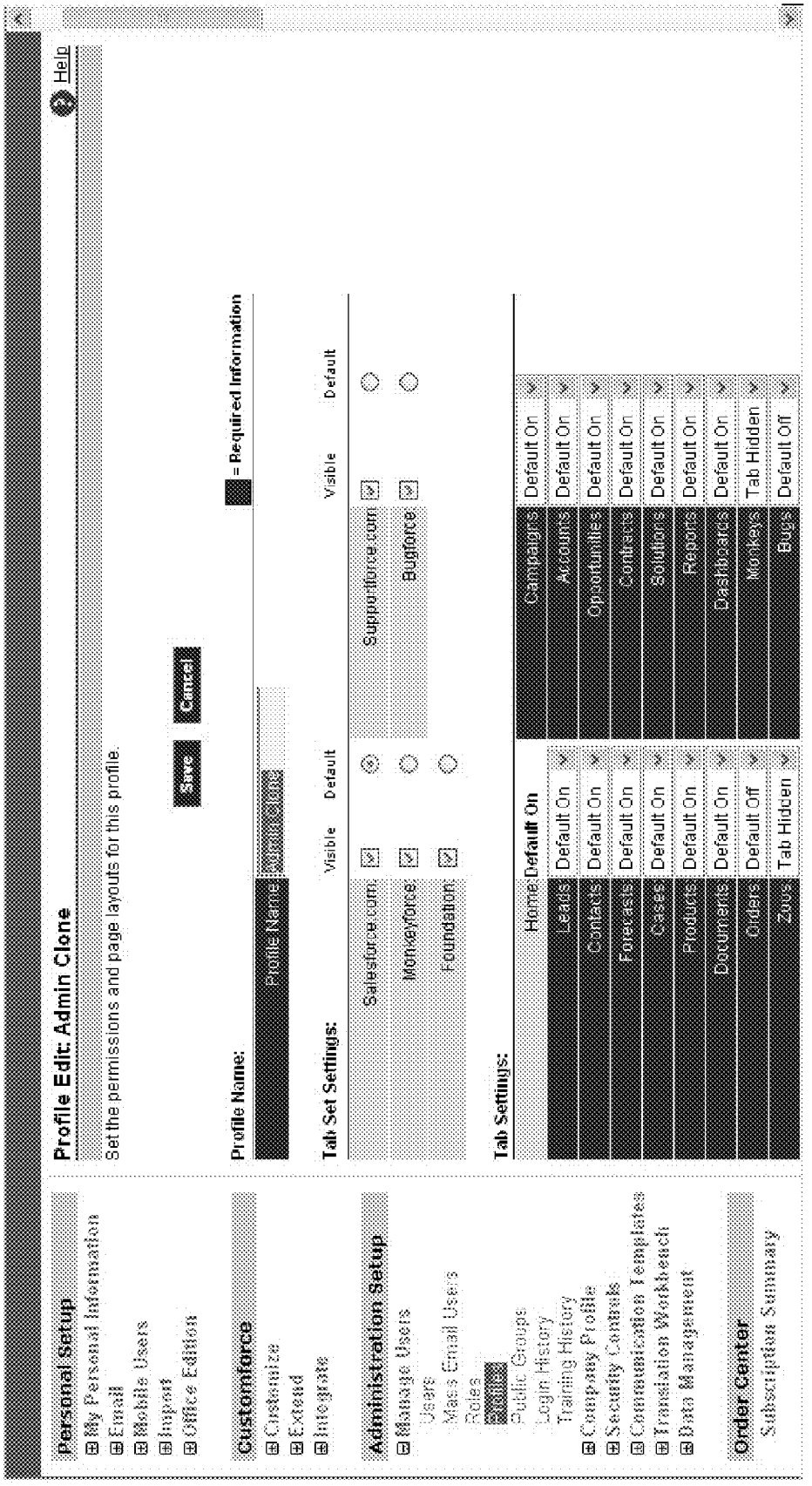
| | | | |
|---|---|---|---|
| Home: | **Default On** | Campaigns: | **Default On** |
| Leads: | **Default On** | Accounts: | **Default On** |
| Contacts: | **Default On** | Opportunities: | **Default On** |
| Forecasts: | **Default On** | Contracts: | **Default On** |
| Cases: | **Default On** | Solutions: | **Default On** |
| Products: | **Default On** | Reports: | **Default On** |
| Documents: | **Default On** | Dashboards: | **Default On** |
| Orders: | **Default Off** | Monkeys: | **Tab Hidden** |
| Zoos: | **Tab Hidden** | Bugs: | **Default Off** |
| Releases: | **Default Off** | Volunteer Activities: | **Default Off** |

**Record Type Settings:**

| | | | |
|---|---|---|---|
| Accounts: | | Contacts: | |
| Leads: | | Opportunities: | |
| Cases: | | Campaigns: | |
| Tasks: | | Events: | |
| Contracts: | | Solutions: | |
| Products: | | | |

**Administrative Permissions:**

FIG. 19: Profile Edit Page



# SYSTEMS AND METHODS FOR IMPLEMENTING MULTI-APPLICATION TABS AND TAB SETS

## CROSS-REFERENCES TO RELATED APPLICATIONS

**[0001]** This application is a continuation of U.S. patent application Ser. No. 12/831,072 (Atty Docket No. 021735-001110US) filed Jul. 6, 2010 which is a continuation of U.S. patent application Ser. No. 11/075,546, (Atty Docket No. 021735-001100US) filed Mar. 8, 2005, the contents of which are hereby incorporated by reference in their entirety.

## REFERENCE TO A "SEQUENCE LISTING," A TABLE, OR A COMPUTER PROGRAM LISTING APPENDIX SUBMITTED ON A COMPACT DISK

**[0002]** NOT APPLICABLE

## BACKGROUND OF THE INVENTION

**[0003]** The present invention relates generally to databases, and more particularly to systems and methods for creating and displaying custom tab and tab set objects for use in a multi-tenant and/or multi-application database system.

**[0004]** In multi-tenant database systems, such as the salesforce.com service, a multi-tenant architecture is used wherein customer organizations (i.e., tenants) share applications and data, and database resources in one logical database. The database tables themselves are typically shared. For example, each entity in the data model typically contains an organization_id column that distinguishes rows for each tenant. All queries and data manipulation in the context of a tenant filter on this (indexed) organization_id column to ensure proper security and the appearance of virtual private databases. In the salesforce.com system, for example, this strategy is used to expose standard entities such as Account, Contact, Lead, and Opportunity entities to customers.

**[0005]** However, customers may wish to add their own custom objects and applications to the system in addition to the standard objects and standard applications already provided. In a traditional client/server application, where the customer has its own physical database, adding custom objects is typically done via DDL (data definition language) against that database to create new physical schema—tables and columns. In an online multi-tenant database system, such as the salesforce.com service, this approach may be untenable for various reasons. For example, for a database system with a large population of tenants (e.g., on the order of 1,000 or 10,000 or more tenants), the union of all desired schema would overwhelm the underlying data dictionary catalog (e.g., Oracle dictionary). Additionally, the maintenance of all of these schema objects would be a nearly impossible burden for DBAs (database administrators). Further, current relational databases do not support online DDL (in a highly concurrent transactional system) well enough for organizations to remain logically independent. Specifically, the creation of schema by one organization could lock an application for all other customers causing unacceptable delays.

**[0006]** U.S. patent application Ser. No. 10/817,161, entitled "Custom Entities and Fields in a Multi-Tenant Database System" filed on Apr. 2, 2004, which is herein incorporated by reference in its entirety, discloses systems and methods for creating and customizing objects such as entities and fields. The systems and methods presented therein offer a flexible approach to storing variable schema data in a fixed physical schema so as to overcome the above and other problems.

**[0007]** However, in addition to providing the ability to create custom database objects such as entities and fields, it would be desirable to provide a sharing model that allows users to easily and conveniently share such standard and custom objects and applications among users or groups of users. The sharing model should allow users to create custom views that provide user access to various objects and application elements. Such sharing model should also provide individual users the ability to customize individual views of the model so as to provide a more convenient, user-friendly database and application user interface environment to the user.

## BRIEF SUMMARY OF THE INVENTION

**[0008]** The present invention provides a novel sharing model that enables users to easily and conveniently create and share standard and custom database objects, and applications, among users or groups of users.

**[0009]** According to the present invention, the sharing model provides systems and methods that allow users to create and customize tabs and tab sets that define relationships between custom objects and fields, standard objects and fields, and applications. A tab visually represents a user interface into an element of an application or into a database object. Selection of a tab provides a user access to the object or element of the application represented by the tab. A tab set is a group of related tabs that work as a unit to provide application functionality. New tabs and tab sets may be defined and tab set views may be customized so that an end user can easily and conveniently switch between the various objects and application elements represented by the displayed tabs and tab set views. In this manner, tabs and tab sets may be used as a means for a user to switch between objects and between applications in a multiple application environment, such as an on-demand web-based hosted application environment. Further, using profiles, a creator of a tab set or other user with appropriate permission may define which users may see and use a tab set definition, and therefore which users are allowed access to the custom objects and applications to which tabs in the tab set definition provide access.

**[0010]** Reference to the remaining portions of the specification, including the drawings and claims, will realize other features and advantages of the present invention. Further features and advantages of the present invention, as well as the structure and operation of various embodiments of the present invention, are described in detail below with respect to the accompanying drawings. In the drawings, like reference numbers indicate identical or functionally similar elements.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0011]** FIG. **1** illustrates an environment wherein a multi-tenant database system might be used.

**[0012]** FIG. **2** illustrates elements of FIG. **1** and various interconnections in more detail.

**[0013]** FIG. **3** is a user interface (UI) view of a standard tab set definition provided by a host system according to one embodiment.

**[0014]** FIG. **4** illustrates various aspects of an example UI including tab sets according to one embodiment.

**[0015]** FIG. **5** shows an example of an "All Tabs" page that supports different modes of viewing tabs according to one aspect.

**[0016]** FIG. **6** shows an example of a page that provides users with the capability to fully customize each tab set, specifying which tabs go in which tab set, and the order of those tabs according to one aspect.

**[0017]** FIG. **7** shows an example of a tab set list view page that allows a user to create and customize tab sets according to one aspect.

**[0018]** FIG. **8** shows an example of a first tab set creation wizard page according to one aspect.

**[0019]** FIG. **9** shows an example of a second tab set creation wizard page that allows a user to provide a custom logo, according to one aspect.

**[0020]** FIG. **10** shows an example of a third tab set creation wizard page that allows a user to specify which tabs belong in the new tab set, and the order they are to be displayed, according to one aspect.

**[0021]** FIG. **11** shows an example of a fourth wizard page that allows a user to specify the tab set's visibility, according to one aspect.

**[0022]** FIG. **12** shows an example of a tab set detail page that shows the details for an existing tab set according to one aspect.

**[0023]** FIG. **13** shows an example of a page presented when a user selects the Delete action according to one aspect.

**[0024]** FIG. **14** shows an example of a tab list view page that allows a user to create and customize tabs according to one aspect.

**[0025]** FIG. **15** shows an example of a first tab creation wizard page according to one aspect.

**[0026]** FIG. **16** shows an example of a second tab creation wizard page according to one aspect.

**[0027]** FIG. **17** shows an example of a wizard page including an additional step for the New action for both Custom Object Tabs and Web Tabs.

**[0028]** FIG. **18** shows an example of a Profile Detail Page showing details of the selections as were made in the Profile Edit Page shown in FIG. **19**.

**[0029]** FIG. **19** shows an example of a Profile Edit page according to one aspect.

## DETAILED DESCRIPTION OF THE INVENTION

**[0030]** FIG. **1** illustrates an environment wherein a multi-tenant database system might be used. As illustrated in FIG. **1** (and in more detail in FIG. **2**) any user systems **12** might interact via a network **14** with a multi-tenant database system (MTS) **16**. The users of those user systems **12** might be users in differing capacities and the capacity of a particular user system **12** might be entirely determined by permissions for the current user. For example, where a salesperson is using a particular user system **12** to interact with MTS **16**, that user system has the capacities allotted to that salesperson. However, while an administrator is using that user system to interact with MTS **16**, that user system has the capacities allotted to that administrator. Thus, different users will have different capabilities with regard to accessing and modifying application and database information, including tab and tab set definition and profile information, depending on a user's permission level.

**[0031]** Network **14** can be a LAN (local area network), WAN (wide area network), wireless network, point-to-point network, star network, token ring network, hub network, or other configuration. As the most common type of network in current use is a TCP/IP (Transfer Control Protocol and Internet Protocol) network such as the global internetwork of networks often referred to as the "Internet" with a capital "I," that will be used in many of the examples herein. However, it should be understood that the networks that the present invention might use are not so limited, although TCP/IP is the currently preferred protocol.

**[0032]** User systems **12** might communicate with MTS **16** using TCP/IP and, at a higher network level, use other common Internet protocols to communicate, such as HTTP, FTP, AFS, WAP, etc. As an example, where HTTP is used, user system **12** might include an HTTP client commonly referred to as a "browser" for sending and receiving HTTP messages from an HTTP server at MTS **16**. Such HTTP server might be implemented as the sole network interface between MTS **16** and network **14**, but other techniques might be used as well or instead. In some implementations, the interface between MTS **16** and network **14** includes load sharing functionality, such as round-robin HTTP request distributors to balance loads and distribute incoming HTTP requests evenly over a plurality of servers. Preferably, each of the plurality of servers has access to the MTS's data, at least as for the users that are accessing that server.

**[0033]** In preferred aspects, the system shown in FIG. **1** implements a web-based customer relationship management (CRM) system. For example, in one aspect, MTS **16** can include application servers configured to implement and execute CRM software applications as well as provide related data, code, forms, web pages and other information to and from user systems **12** and to store to, and retrieve from, a database system related data, objects and web page content. With a multi-tenant system, tenant data is preferably arranged so that data of one tenant is kept separate from that of other tenants so that one tenant does not have access to another's data, unless such data is expressly shared. It should, however, be appreciated that system **16** can also implement applications other than, or in addition to, a CRM application. For example, system **16** may provide tenant access to multiple hosted applications including a CRM application.

**[0034]** One arrangement for elements of MTS **16** is shown in FIG. **1**, including a network interface **20**, storage **22** for tenant data, storage **24** for system data accessible to MTS **16** and possibly multiple tenants, program code **26** for implementing various functions of MTS **16**, and a process space **28** for executing MTS system processes and tenant-specific processes, such as running applications as part of an application hosting service.

**[0035]** Several elements in the system shown in FIG. **1** include conventional, well-known elements that need not be explained in detail here. For example, each user system **12** could include a desktop personal computer, workstation, laptop, PDA, cell phone, or any WAP-enabled device or any other computing device capable of interfacing directly or indirectly to the Internet or other network connection. User system **12** typically runs an HTTP client, e.g., a browsing program, such as Microsoft's Internet Explorer™ browser, Netscape's Navigator™ browser, Opera's browser, or a WAP-enabled browser in the case of a cell phone, PDA or other wireless device, or the like, allowing a user (e.g., subscriber of a CRM system) of user system **12** to access, process and view information, pages and applications available to it from MTS **16** over network **14**. Each user system **12** also typically includes one or more user interface devices, such as a key-

board, a mouse, touch screen, pen or the like, for interacting with a graphical user interface (GUI) provided by the browser on a display (e.g., monitor screen, LCD display, etc.) in conjunction with pages, forms, applications and other information provided by MTS **16** or other systems or servers. For example, the user interface device can be used to select tabs and tab sets and otherwise allow a user to interact with the various GUI pages as described herein.

**[0036]** As discussed above, the present invention is suitable for use with the Internet, which refers to a specific global internetwork of networks. However, it should be understood that other networks can be used instead of the Internet, such as an intranet, an extranet, a virtual private network (VPN), a non-TCP/IP based network, any LAN or WAN or the like.

**[0037]** According to one embodiment, each user system **12** and all of its components are operator configurable using applications, such as a browser, including computer code run using a central processing unit such as an Intel Pentium processor or the like. Similarly, MTS **16** (and additional instances of MTS's, where more than one is present) and all of their components might be operator configurable using application(s) including computer code run using a central processing unit such as an Intel Pentium processor or the like, or multiple processor units. Computer code for operating and configuring MTS **16** to intercommunicate and to process web pages, applications and other data and media content as described herein is preferably downloaded and stored on a hard disk, but the entire program code, or portions thereof, may also be stored in any other volatile or non-volatile memory medium or device as is well known, such as a ROM or RAM, or provided on any media capable of storing program code, such as a compact disk (CD) medium, digital versatile disk (DVD) medium, a floppy disk, and the like. Additionally, the entire program code, or portions thereof, may be transmitted and downloaded from a software source, e.g., over the Internet, or from another server, as is well known, or transmitted over any other conventional network connection as is well known (e.g., extranet, VPN, LAN, etc.) using any communication medium and protocols (e.g., TCP/IP, HTTP, HTTPS, Ethernet, etc.) as are well known. It will also be appreciated that computer code for implementing aspects of the present invention can be implemented in any programming language that can be executed on a server or server system such as, for example, in C, C++, HTML, any other markup language, Java, JavaScript, any other scripting language such as VBScript, and many other programming languages as are well known.

**[0038]** According to one embodiment, each MTS **16** is configured to provide web pages, forms, applications, data and media content to user systems **12** to support the access by user systems **12** as tenants of MTS **16**. As such, MTS **16** provides security mechanisms to keep each tenant's data separate unless the data is shared. If more than one MTS is used, they may be located in close proximity to one another (e.g., in a server farm located in a single building or campus), or they may be distributed at locations remote from one another (e.g., one or more servers located in city A and one or more servers located in city B). As used herein, each MTS could include one or more logically and/or physically connected servers distributed locally or across one or more geographic locations. Additionally, the term "server" is meant to include a computer system, including processing hardware and process space(s), and an associated storage system and database application (e.g., OODBMS or RDBMS) as is well known in the art. It should also be understood that "server system" and "server" are often used interchangeably herein. Similarly, the databases described herein can be implemented as single databases, a distributed database, a collection of distributed databases, a database with redundant online or offline backups or other redundancies, etc., and might include a distributed database or storage network and associated processing intelligence.

**[0039]** FIG. **2** illustrates elements of MTS **16** and various interconnections in more detail. In this example, the network interface is implemented as one or more HTTP application servers **100**. Also shown is system process space **102** including individual tenant process spaces **104**, a system database **106**, tenant database(s) **108** and a tenant management process space **110**. Tenant database **108** might be divided into individual tenant storage areas **112**, which can be either a physical arrangement or a logical arrangement. Within each tenant storage area **112**, user storage **114** might similarly be allocated for each user.

**[0040]** It should also be understood that each application server **100** may be communicably coupled to database systems, e.g., system database **106** and tenant database(s) **108**, via a different network connection. For example, one server $\mathbf{100}_1$ might be coupled via the Internet **14**, another server $\mathbf{100}_{N-1}$ might be coupled via a direct network link, and another server $\mathbf{100}_N$ might be coupled by yet a different network connection. Transfer Control Protocol and Internet Protocol (TCP/IP) are preferred protocols for communicating between servers **100** and the database system, however, it will be apparent to one skilled in the art that other transport protocols may be used to optimize the system depending on the network interconnect used.

**[0041]** In preferred aspects, each application server **100** is configured to handle requests for any user/organization. Because it is desirable to be able to add and remove application servers from the server pool at any time for any reason, there is preferably no server affinity for a user and/or organization to a specific application server **100**. In one embodiment, therefore, an interface system (not shown) implementing a load balancing function (e.g., an F5 Big-IP load balancer) is communicably coupled between the servers **100** and the user systems **12** to distribute requests to the servers **100**. In one aspect, the load balancer uses a least connections algorithm to route user requests to the servers **100**. Other examples of load balancing algorithms, such as round robin and observed response time, also can be used. For example, in certain aspects, three consecutive requests from the same user could hit three different servers, and three requests from different users could hit the same server. In this manner, MTS **16** is multi-tenant, wherein MTS **16** handles storage of, and access to, different objects, data and applications across disparate users and organizations.

**[0042]** As an example of storage, one tenant might be a company that employs a sales force where each salesperson uses MTS **16** to manage their sales process. Thus, a user might maintain contact data, leads data, customer follow-up data, performance data, goals and progress data, etc., all applicable to that user's personal sales process (e.g., in tenant database **108**). In the preferred MTS arrangement, since all of this data and the applications to access, view, modify, report, transmit, calculate, etc., can be maintained and accessed by a user system having nothing more than network access, the user can manage his or her sales efforts and cycles from any of many different user systems. For example, if a salesperson

is visiting a customer and the customer has Internet access in their lobby, the salesperson can obtain critical updates as to that customer while waiting for the customer to arrive in the lobby.

**[0043]** While each user 's sales data might be separate from other users' sales data regardless of the employers of each user, some data might be organization-wide data shared or accessible by a plurality of users or all of the sales force for a given organization that is a tenant. Thus, there might be some data structures managed by MTS **16** that are allocated at the tenant level while other data structures might be managed at the user level. Because an MTS might support multiple tenants including possible competitors, the MTS should have security protocols that keep data, applications and application use separate. Also, because many tenants will opt for access to an MTS rather than maintain their own system, redundancy, up-time and backup are more critical functions and need to be implemented in the MTS.

**[0044]** In addition to user-specific data and tenant-specific data, MTS **16** might also maintain system level data usable by multiple tenants or other data. Such system level data might include industry reports, news, postings, and the like that are sharable among tenants.

**[0045]** In certain aspects, client systems **12** communicate with application servers **100** to request and update system-level and tenant-level data from MTS **16** that may require one or more queries to database system **106** and/or database system **108**. MTS **16** (e.g., an application server **100** in MTS **16**) generates automatically one or more SQL statements (the SQL query) designed to access the desired information.

**[0046]** Each database can generally be viewed as a collection of objects, such as a set of logical tables, containing data fitted into predefined categories. A "table" is one representation of a data object, and is used herein to simplify the conceptual description of objects and custom objects according to the present invention. It should be understood that "table" and "object" may be used interchangeably herein. Each table generally contains one or more data categories logically arranged as columns or fields in a viewable schema. Each row or record of a table contains an instance of data for each category defined by the fields. For example, a CRM database may include a table that describes a customer with fields for basic contact information such as name, address, phone number, fax number, etc. Another table might describe a purchase order, including fields for information such as customer, product, sale price, date, etc. In some multi-tenant database systems, standard entity tables might be provided for use by all tenants. For CRM database applications, such standard entities might include tables for Account, Contact, Lead and Opportunity data, each containing pre-defined fields.

**[0047]** According to one aspect, a user can design their own custom applications including custom objects, custom tabs, custom fields, and custom page layouts. U.S. patent application Ser. No. 10/817,161, entitled "Custom Entities and Fields in a Multi-Tenant Database System" filed on Apr. 2, 2004, previously incorporated by reference in its entirety, discloses systems and methods for creating and customizing objects such as entities and fields. Using the techniques of the present invention, tabs and tab sets can be created and customized to define relationships between custom objects and fields, standard objects and fields, and applications and to track related data.

Custom Tabs and Tab sets

**[0048]** According to the present invention, in general, a tab represents a user interface into an element of an application or into a database object. Selection of a tab provides a user access to the object or element of the application represented by the tab. A tab set is a group of related tabs that work as a unit to provide application functionality. The present invention allows for new tabs and tab sets to be defined and tab set views to be customized so that an end user can easily and conveniently switch between the various objects and application elements represented by the displayed tabs and tab sets. In one aspect, for example, tabs and tab sets may be used as a means to switch between applications in a multiple application environment, such as an on-demand web-based hosted application environment.

**[0049]** In one embodiment, a tab set includes a name, a logo, and an ordered list of tabs. A tab set is typically viewed in a graphical user interface (GUI) environment, e.g., using a browser application running on a user's computer system. For example, FIG. **3** illustrates an example of a portion of a GUI view of a tab set. As shown in FIG. **3**, the currently selected tab set **100** includes a logo **110** and a tab set selector **120**, which preferably is only included when multiple tab sets are available for selection by a user. The tab set selector **120** in certain aspects includes a dropdown list, or picklist, that is available when multiple tab sets are provided and that is used to switch between the available sets of tabs. When a user changes the selection, the displayed tabs, order of tabs and "new xxx" bar are changed based on the selected tab set definition.

**[0050]** The tab set shown in FIG. **3** represents a standard tab set definition provided by a host system, e.g., MTS **16**. Standard tab sets are pre-defined sets of tabs, e.g., imported from a source that provides a capability (e.g., templating capability) that determines which tabs, tab sets and data a tenant or user is initially provisioned with. One example of standard tab sets are provided by the salesforce.com website through its subscription CRM service. Using these standard tab sets, users are provided access to standard tables or entities such as Account, Contact, Lead and Opportunity entities. For example, tabs representing the salesforce.com Account, Contact, Lead and Opportunity entities and others are shown in FIG. **3**. As another example, salesforce.com also provides the ability to create custom entities as well as custom fields for standard entities. Using the present invention, a user can create a tab set including tabs representing custom entities and fields.

**[0051]** According to one aspect, a user may create custom tab sets and custom tabs. Preferably only administrator level users are provided with tab set creation functionality based on their stored permissions. Additionally, users may customize their view of tab sets, including the order of displayed tabs and which tabs in a tab set are displayed. To allow users to conveniently organize their tabs, each tab may appear in any and all tab sets if desired. Preferably, any user can edit tab combination and order, but cannot rename or replace a logo; tab set naming and logo selection are preferably only administrator level functions. For example, administrators may create new tab sets and customize existing tab sets. For all tab sets, an administrator can specify which tabs are included, and the order that the tabs should be displayed. For organization-specific tab sets, an administrator can also specify the name and provide an optional logo. For standard tab sets provided by the host system, e.g., tab sets provided by sales-

force.com, such as Salesforce.com and Supportforce.com tab sets, an administrator preferably cannot change the name or logo, nor can the administrator delete the standard tab set. Preferably, any user can fully customize their view of all the tab sets they have permission to view. For users who only have access to a single tab set, in one aspect these users will not see the tab set selector **120**. As will be discussed below, the tabs a user can see are based on the user's permission level. A profile for each tab set allows an administrator level user to set the profile level viewability of tabs and tab sets, e.g., so that groups of users at certain permission levels may be restricted from viewing certain tabs or tab sets, and therefore also may be restricted from accessing or viewing certain objects and applications referenced by the restricted tabs or tab sets.

**[0052]** Thus, in one aspect, a tab set can be thought of as a filter that is overlaid on top of an existing profile-level tab visibility definition. An administrator sets the default tabs that are included in each tab set filter, but each user can override as they like—the only thing they preferably cannot change is the tab set name and logo. The net result is that tab sets are quite lightweight and flexible. A particular meaning to a tab set is not enforced; each user can generally use tab sets as they wish.

Tab Set User Interface

**[0053]** FIG. **4** illustrates various aspects of an example UI including tab sets according to one embodiment. As shown in FIG. **4**, a portion of the screen (e.g., the upper-right-hand corner of each screen where Setup, Help and Training, Logout and other links are typically provided) may be shaded or otherwise prominently displayed to indicate that this section applies (e.g., is persistent) across all applications. For those users that have permission to view more than one tab set, the tab set selector **120** is also shown in this portion. As shown, the selector **120** includes an "S" logo, which in this example is used to brand the salesforce.com on-demand operating system, and a picklist **122**, listing each of the tab sets the user has permission to view. When the user selects a tab set from the picklist **122**, as shown in FIG. **4***b*, the tab set's logo is prominently displayed, e.g., in the upper-left-hand corner of the page as shown. If the tab set is not supplied by the host system, e.g., in this example the host is salesforce.com, and the customer provides a logo (i.e., the tab set does not use a default host logo), then an additional host-supplied graphic, such as a "powered by Customforce.com" graphic, may be displayed next to the logon on the right-hand-side of the banner, as shown in FIG. **4***c*.

**[0054]** The tabs (except the Home tab) are replaced with the selected tab set's tabs, in the order specified by the stored tab set definition. By default, the Home tab is displayed as the current tab. The "new" bar is also replaced with a list of links for creating objects related to those objects referenced by the tabs in the selected tab set. For example, as shown in FIG. **4***d*, if the tab set includes the following tabs: Zoos and Monkeys, then the "new" bar would have the following: New Zoo, New Monkey and Recycle Bin. The order of links in the "new" bar preferably matches the order of tabs in the tab set, but it is not necessary as can be seen in FIG. **4***c*.

**[0055]** In one aspect, to help users understand the tab set selector **120**, when the mouse hovers over either the "S" logo or the picklist **122**, a tool tip will be displayed explaining the concept of tab sets.

**[0056]** If the user creates a new object or views an existing object, and that object is not referenced in the selected tab set, then the behavior of the system is preferably unaltered, e.g., the same as that which occurs for viewing or creating new objects that do not have a tab. Even if that object has a tab in another of the user's tab sets, the selected tab set preferably does not change. Where a recent items list is used, the list is preferably not filtered by the selected tab set definition. If the user clicks an item that does not have a tab displayed in the current tab set, its detail page will be displayed but its tab will not be displayed.

**[0057]** FIG. **5** shows an example of a page ("All Tabs" Page) that supports different modes of viewing tabs according to one aspect. The mode is selected using the "View" dropdown list (picklist). In one mode, all tabs the user has access to see are displayed in alphabetical order by selecting "All Tabs". In another mode, the tabs are grouped by tab set, in display order. As shown, the tabs for a standard salesforce.com tab set are shown in display order. The dropdown list allows the user to pick which tab set to display.

**[0058]** FIG. **6** shows an example of a page that provides users with the capability to fully customize each tab set, specifying which tabs go in which tab set, and the order of those tabs. The tabs listed in the dueling listboxes include all tabs that are not hidden for the user's profile. As shown in FIG. **6**, the user may specify which tabs belong in a selected tab set using the "add" and "Remove" buttons. The user may also rearrange the order in which those tabs should appear using the "up" and "Down" buttons. Users are preferably only be able to view those tabs that they have permission to see, regardless of whether or not the tab is listed in a tab set.

**[0059]** FIG. **7** shows an example of a tab set list view page that allows a user to create and customize tab sets according to one aspect. Any user that has the appropriate permissions may administer tab sets, e.g., create, delete, edit, etc. The list view preferably includes a visual table containing an entry for each available tab set as shown. Each entry shows the tab set name and a description. As shown, the list view page supports a New action to create a new custom tab set by selecting a "New" button **125**, an Edit action to edit a tab set by selecting an edit selector **126**, and a Delete action to delete a tab set by selecting a delete selector **127**.

**[0060]** Selection of the New button **125** invokes a tab set creation "wizard". FIG. **8** shows an example of a first wizard page according to one aspect. The wizard provides entry boxes for the user to enter the name (label) of the tab set as well as an optional description. The tab set name size is preferably limited, e.g., to a 40 character max length. Custom tab sets preferably cannot be named after standard tab sets provided by a host system. For example, in the case where salesforce.com provides the host system, "Salesforce", "Supportforce" and "Customforce" (with or without ".com") may not be used. The tab set name also preferably cannot contain HTML characters.

**[0061]** FIG. **9** shows an example of a second wizard page that allows a user to provide a custom logo, according to one aspect. The second page is invoked when information in the first page has been entered and the user selects "next". As shown in FIG. **9**, the user may supply a logo for the new tab set (by default, a host system logo such as the Customforce.com logo is pre-populated). After an image has been identified or uploaded to the system, the image may be scaled down or clipped if it is larger than a predetermined size (e.g., width: 235 pixels; height: 55 pixels). Scaling will maintain the origi-

nal logo's aspect ratio. Although .gif images are preferred, other image formats such as JPEG, Bitmap and others may be used.

**[0062]** FIG. **10** shows an example of a third wizard page that allows a user to specify which tabs belong in the new tab set, and the order they are to be displayed, according to one aspect. The third page is invoked when the user selects "save" in the second page. As shown in FIG. **10**, the user may specify which tabs belong in this tab set using the "add" and "Remove" buttons. The user may also rearrange the order in which those tabs should appear using the "up" and "Down" buttons. Users are preferably only be able to view those tabs that they have permission to see, regardless of whether or not the tab is listed in a tab set.

**[0063]** FIG. **11** shows an example of a fourth wizard page that allows a user to specify the tab set's visibility, according to one aspect. The fourth page is invoked when the user selects "save" in the third page. As shown in FIG. **11**, the user may choose the user profiles for which the new custom tab set will be available and to choose whether this custom tab set will be a default tab set. In one aspect, the user is able to specify the tab set's visibility and default behavior for each User Profile using a series of selectable checkboxes as shown.

**[0064]** FIG. **12** shows an example of a tab set detail page that shows the details for an existing tab set. Detail information may include the label, description, logo, included tabs, "created by" information and "last updated" information. This page may be invoked, for example, by selecting the tab set label from the All Tabs page shown in FIG. **7**. As shown in FIG. **12**, the user may also select the Edit action from this page.

**[0065]** Selection of the Edit action, e.g., in FIG. **7** or FIG. **12**, invokes a wizard that is nearly identical to the wizard invoked by the New action as described above. However, detail entering and logo selection steps are not available for standard tab sets supplied by the host system, e.g., salesforce.com. Also, the last step of the wizard preferably includes a check box allowing the user to specify whether or not any changes should override any user's personal customizations.

**[0066]** FIG. **13** shows an example of a page presented when a user selects the Delete action. When selected, the user is asked which other tab set should be set as the default tab set for those users that had the tab set that is about to be deleted selected as the default tab set. Because tab sets are effectively filters, they can be deleted without affecting the profile tab visibility preferences. The order of tabs preferably reverts to the current pre-tab set order scheme (i.e., standard tabs in hard-coded order, custom tabs in creation order).

**[0067]** According to one aspect, a user can also create custom tabs. FIG. **14** shows an example of a tab list view page that allows a user to create and customize tabs according to one aspect. The list view shown preferably includes a visual table containing an entry for each available tab as shown. Each entry shows the tab name, a tab style and a description. As shown, the list view page supports a New action to create a new custom tab by selecting a "New" button, an Edit action to edit an existing tab by selecting an edit selector, and a Delete action to delete an existing tab by selecting a delete selector. Again, preferably only administrator level users should have access to administer tabs. A user can create custom object tabs that reference custom objects, and web tabs that allow users to embed external web applications and content within a window.

**[0068]** Selection of the New button invokes a tab creation "wizard". FIG. **15** shows an example of a first tab creation wizard page according to one aspect. This wizard provides entry boxes for the user to select the name of an existing custom object (or select to create a new custom object) as well as a tab style and an optional tab description.

**[0069]** FIG. **16** shows an example of a second tab creation wizard page according to one aspect. The second page is invoked when information in the first page has been selected or entered and the user selects "next". The user may choose the user profiles for which the new custom tab will be available. In one aspect, the user is able to specify the tab's visibility and default behavior for each User Profile using a series of selectable checkboxes as shown.

**[0070]** According to one aspect, an additional last step is added to the wizards for the New action for both Custom Object Tabs and Web Tabs, as shown in FIG. **17**. In this step, the user specifies in which tab sets the new tab is visible, and whether or not this new tab should be appended to users' existing personal customizations.

**[0071]** In one aspect, a new "Tab Set Settings" section is added to each Profile object, listing all the tab sets and specifying, for the profile, which users may see the tab set, and which tab set should be selected the first time a user logs on (from then on, the tab set that is selected when a user logs on is preferably the same as the tab set that was selected when the user last logged off). As shown in the Profile Edit page of FIG. **19**, visibility of a tab set is indicated by a check box (since users may have access to more than one tab set), but which tab set should be initially selected is specified by a radio button (there may only be one default tab set). FIG. **18** shows an example of a Profile Detail Page showing details of the selections as were made in the Profile Edit Page of FIG. **19**.

Server Side Operations and Data Model Definitions

**[0072]** Logo

**[0073]** In one aspect, a logo part can be overridden by a user's tab set definition. An image provided by a user is preferably stored "in document", e.g., in a document storage module or facility that allows customers to store any document types they like. For example, the logo is stored as a document, which is then referenced by the wizard used when creating a new tab set. An HTML IMG tag is then automatically generated which points to (via a URL) the logo in the document store facility. It is preferred that a restriction be set on the rectangular size of the image, otherwise a user adversely impact the UI by posting large images, e.g., huge .gif files. Therefore, in one aspect, depending on the size of the image provided, the image may be clipped or scaled. For example, the example below will clip an image if it is bigger than 55 pixels in height.

```
<div style="position:absolute;height:55px; overflow:hidden; ">
    <img border=0 src="xxx.gif">
</div>
```

**[0074]** Tab Selector

**[0075]** In one aspect, the available values in a tab selector are defined in each profile, with a default. Each user belongs to a profile, which is used to help simplify the administration of a group of users. The default tab is used when user logs in to the system. If there is only one tab set in the user's profile,

the user will not see the selector. In certain aspects, session cookies are used to maintain the current selection of the tab selector **120** until the user logs out. Saving a user preference into the database is another way to maintaining the user's selection.

**[0076]** Multi-Browser Operation

**[0077]** Even if a user opens two (or more) browser windows to access the host system, selection of the each tab selector (i.e., tab set selection) is preferably maintained for each browser. In this case, it would be desirable if each browser instance that shares the same session had its own cookie. However, there is no easy way to generate a browser-instance specific cookie id. In one aspect, to resolve this case, every selection change on a tab selector immediately calls an application server to reload. When tab set selection occurs, a session cookie is set and sent back to the server. The session cookie may then be set again to make sure the value is properly passed to the server.

**[0078]** User Preference

**[0079]** A user can edit each tab set they have permission to view, by customizing which tabs are in the tab set, as well as the order of the tabs. Tabs that a user can choose to display are preferably not limited to the current tab set, but rather to all tabs that they can see which are defined in a stored profile. Tabs that are marked as "Hidden" in a user's profile, however, are not selectable.

**[0080]** Design

**[0081]** In-Memory Data Structure

**[0082]** Since the tab set selector and tabs are referred by almost every page, in one aspect, tab set information is preferably cached in the relevant database tables. Required information may include, for example, 1) each tab set (w/ array of tabs), 2) Set of Tab Set (for constructing Tab Set Selector in Profile) and 3) Set of TabSet (subset of master tab set) for user preference.

**[0083]** Tab Set

**[0084]** Each tab set contains a name, description, image and list of tabs, with no default. This information is not specific to a given user, but exists across an entire organization or tenant, and is stored in a database table that is shared across an entire organization (but is not shared across organizations). There are two types of tab sets: standard tab sets and custom tab sets. Standard tab sets are defined by the host system, e.g., salesforce.com, and users are limited in their ability to customize the standard tab sets.

**[0085]** Below is an example of useful Java class definitions and database table definitions to define the Tab Set data structures:

```
class TabSetInfo implements Serializable
    String id;
    int type;        // 0: Custom, >0: Standard
    String masterLabel;
    String documentId;
    String[ ] tabs;
```

**[0086]** In one aspect, the tab sets initially provided to a tenant are defined in a template, and therefore different tenants can start with different tab sets, and those tab sets can be configured with different tabs.

**[0087]** In one aspect, each user is mapped to a "profile", which determines the permissions, including tab set visibility, for all the users in that profile. To manage this information, a relation is defined as a list of TabSets with a default flag and is held in a database table:

```
ProfileInfo:
    String[ ] tabSetIds;    // array of TabSetInfo IDs.
    int defaultTabSetIndex; // index in tabSetIds
```

**[0088]** Because users can also customize their tab sets, the user's customizations are stored in a database table in one aspect.

**[0089]** API Considerations

**[0090]** In addition to making tab sets available via a web browser, in certain aspects tab sets are also available via a Web Services (SOAP/HTTP) API, and can therefore be implemented via other client software, including an offline browser, a wireless client, etc. When translated to Java, one example of a useful API is as follows:

```
public DescribeTabSetResult[ ] describeTabs( ) throws
RemoteException, UnexpectedErrorFault;
```

**[0091]** Note that the result is an array of DescribeTabSetResult objects, which are defined, when translated to Java, as follows:

```
public class DescribeTabSetResult    implements java.io.Serializable {
    private java.lang.String label;
    private java.lang.String logoUrl;
    private boolean selected;
    private com.sforce.soap.partner.DescribeTab[ ] tabs;
    public DescribeTabSetResult( ) {
    }
    public java.lang.String getLabel( ) {
        return label;
    }
    public java.lang.String getLogoUrl( ) {
        return logoUrl;
    }
    public boolean isSelected( ) {
        return selected;
    }
    public com.sforce.soap.partner.DescribeTab[ ] getTabs( ) {
        return tabs;
    }
    public void setTabs(com.sforce.soap.partner.DescribeTab[ ] tabs) {
        this.tabs = tabs;
    }
};
```

**[0092]** Finally, this class includes references to a list of DescribeTab objects, which can be defined in Java as follows:

```
public class DescribeTab    implements java.io.Serializable {
    private boolean custom;
    private java.lang.String label;
    private java.lang.String sobjectName;
    private java.lang.String url;
    public DescribeTab( ) {
    }
    public boolean isCustom( ) {
        return custom;
```

-continued

```
        }
        public java.lang.String getLabel( ) {
            return label;
        }
        public java.lang.String getSobjectName( ) {
            return sobjectName;
        }
        public java.lang.String getUrl( ) {
            return url;
        }
    };
```

**[0093]** While the invention has been described by way of example and in terms of the specific embodiments, it is to be understood that the invention is not limited to the disclosed embodiments. To the contrary, it is intended to cover various modifications and similar arrangements as would be apparent to those skilled in the art. Therefore, the scope of the appended claims should be accorded the broadest interpretation so as to encompass all such modifications and similar arrangements.

What is claimed is:

**1**. A computer-based method for providing an interface, comprising:

providing, to an end user of a user system, a list of a plurality of tabs that are available for selection by a user, wherein the plurality of tabs is selected from a group including tabs for accessing at least one element of each of at least two applications;

receiving, from the user system, one or more of said plurality of tabs; and

storing, as at least a portion of the interface, at least one tabset definition defining a tabset corresponding to the one or more of said plurality of tabs.

**2**. A computer-based method according to claim **1**, wherein:

at least one of the tabs in the list of a plurality of tabs provides for accessing an element of a first application; and

at least one of the tabs in the list of a plurality of tabs provides for accessing an element of a second application.

**3**. A computer-based method according to claim **1**, wherein the list of a plurality of tabs is selected from a group that further includes tabs for accessing one or more data objects.

**4**. A computer-based method according to claim **3**, wherein the data object includes a database object.

**5**. A computer-based method according to claim **4**, wherein the database object is selected from a group including tables, entities and objects.

**6**. A computer-based method according to claim **1**, wherein one or more of the tabset and at least one of the plurality of tabs provides for switching between the applications.

**7**. A computer-based method according to claim **1**, wherein the applications are selected from a group including standard applications provided by a multiple-application host, and custom applications created by one or more users of the multi-application host and shared with other multiple-application host users.

**8**. A computer-based method according to claim **3**, wherein the at least one of the data objects is selected from a group including standard data objects provided by an multiple-application host, and custom data objects created by one or more users of the multi-application host and shared with other multiple-application host users.

**9**. A computer-based method according to claim **8**, wherein the custom data objects include standard data objects that are customized by a user.

**10**. A computer-based method according to claim **3**, wherein one or more of the tabset and at least one of the plurality of tabs provides for switching between the one or more data objects.

**11**. A computer-based method according to claim **1**, further comprising defining, by respective ones of the user of a multiple application host, users that can see a corresponding tabset.

**12**. A computer-based method according to claim **1**, further comprising defining, by respective ones of a multiple application host and a tabset creating user, users that can see a corresponding tab when the interface is presented.

**13**. A computer-based method according to claim **3**, further comprising:

determining whether to provide access by a user to an applications, an application element or a data object according to a capacity of a user system used by the user.

**14**. A computer-based method according to claim **13**, further comprising determining the capacity of the user system according to a permission of the user system user.

**15**. A computer-based system, comprising:

means for providing, to an end user of a user system, a list of a plurality of tabs that are available for selection by a user, wherein the plurality of tabs is selected from a group including tabs for accessing at least one element of each of at least two applications;

means for receiving, from the user system, one or more of said plurality of tabs; and

means for storing, as at least a portion of the interface, At least one tabset definition defining a tabset corresponding to the one or more of said plurality of tabs.

**16**. A computer-based system according to claim **15**, wherein:

at least one of the tabs in the list of a plurality of tabs provides for accessing an element of a first application; and

at least one of the tabs in the list of a plurality of tabs provides for accessing an element of a second application.

**17**. A computer-based system according to claim **15**, wherein the list of a plurality of tabs is selected from a group that further includes tabs for accessing one or more data objects.

**18**. A computer-based system according to claim **15**, wherein the data object includes a database object.

**19**. A computer-based system according to claim **15**, wherein one or more of the tabset and at least one of the plurality of tabs provides for switching between the applications.

**20**. A computer-based system according to claim **18**, wherein:

the applications are selected from a group including standard applications provided by an multiple-application host, and custom applications created by one or more users of the multi-application host and shared with other multiple-application host users; and

the database objects are selected from a group including standard database objects provided by an multiple-application host, and custom database objects created by one or more users of the multi-application host and shared with other multiple-application host users.

**21**. A computer readable storage medium storing program code for causing a computer to perform the steps of:

providing, to an end user of a user system, a list of a plurality of tabs that are available for selection by a user, wherein the plurality of tabs is selected from a group including tabs for accessing at least one element of each of at least two applications;

receiving, from the user system, one or more of said plurality of tabs; and

storing, as at least a portion of the interface, At least one tabset definition defining a tabset corresponding to the one or more of said plurality of tabs.

* * * * *