Content Management System

CIS 726

Kansas State University

Department of Computing and Information Sciences

Manhattan, KS.

Narayanan Raghavan

Srivatsan Ravikumar

1.0 Introduction

The growth in web technologies and the need for improved and more secured role based web management has spurred the growth for content management systems. A content management system, as the name implies, is managing web content and, look and feel via role based access for viewing content, editing content and administering the portal dynamically. This role based security permits organizations to create a hierarchical multi-tier architecture model for securely managing individual pages or even individual modules on a page.

1.1 Why a Portal?

The basic idea behind a portal is to provide an easy interface to users while maintaining the flexibility for developers to plug-in new features. It is a tool that allows people to communicate interactively and effectively by sharing documents, images and latest developments almost instantly. It provides a framework that works on top of role-based security. This can be adapted to develop any site that requires content management. Hence, it caters to the needs of a wide range of audience.

2.0 Architecture

While portals look like any other static web site with tabs to navigate between pages and page content organized in a neat readable format, they provide a dynamic way to modify, maintain and monitor usage. The basic architecture of the portal consists of tabs and modules. A collection of modules on a page constitutes the contents of the tab and a collection of tabs make the entire portal. We will look individually into each element that constitutes the portal.

2.1 Portal Tabs

Tabs in the portal constitute the basic navigation bar. From a user's point of view this is the basic container of all data and hence its accessibility is of at most importance. Controlling the visibility of the tabs via a role-based mechanism allows managing users belonging to a wide spectrum. Users with different roles only see the tabs that they are authorized to view. Thus, the portal can be set up in such a way that two users with different roles see completely different web contents. Portal administrators manage tabs and they can dynamically create new tabs and add new modules onto the tab.

2.2 Portal Modules

Modules act as the container of all data and content. Modules in a tab can be either in the right pane or left pane depending on where the content is to be displayed. This way we use the entire space on the computer screen. Like view permissions on tabs, role based security can also be used to control edit permission on modules. Hence, while one user can only view the contents in the

module, another user may be given access to modify and maintain the contents on the same module. Module managers manage modules and they can dynamically add or edit content. Administrators can also manage modules but usually they can delegate work to module managers thereby making use of the multi-tier portal architecture.

The portal as a whole can be thought of as a single page containing modules or user controls and their respective editable user interfaces as an arm that extends this central page as shown in figure 1. Navigation tabs only change the modules displayed.

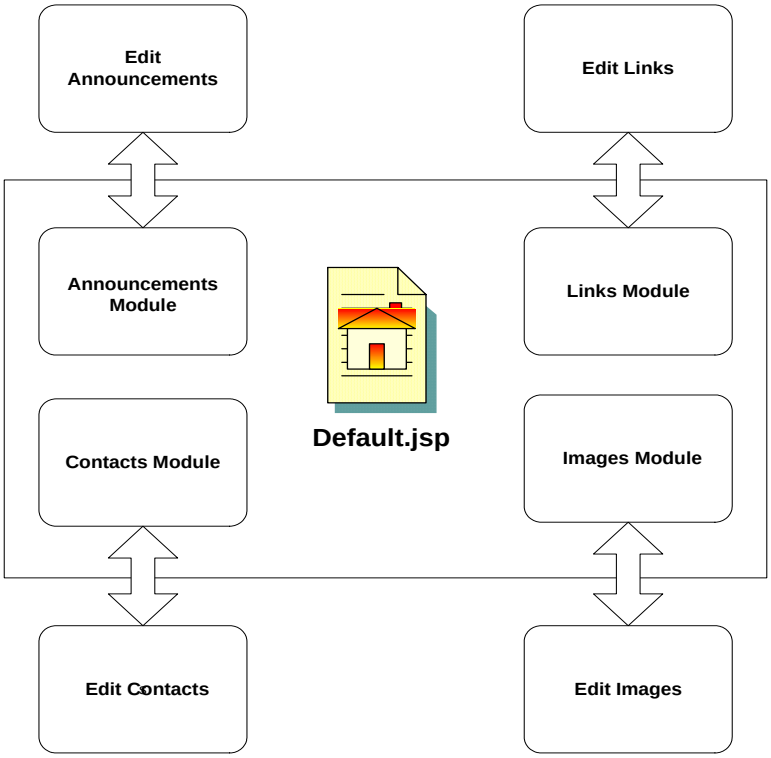


Figure1: The underlying structure of the portal with modules plugged in.

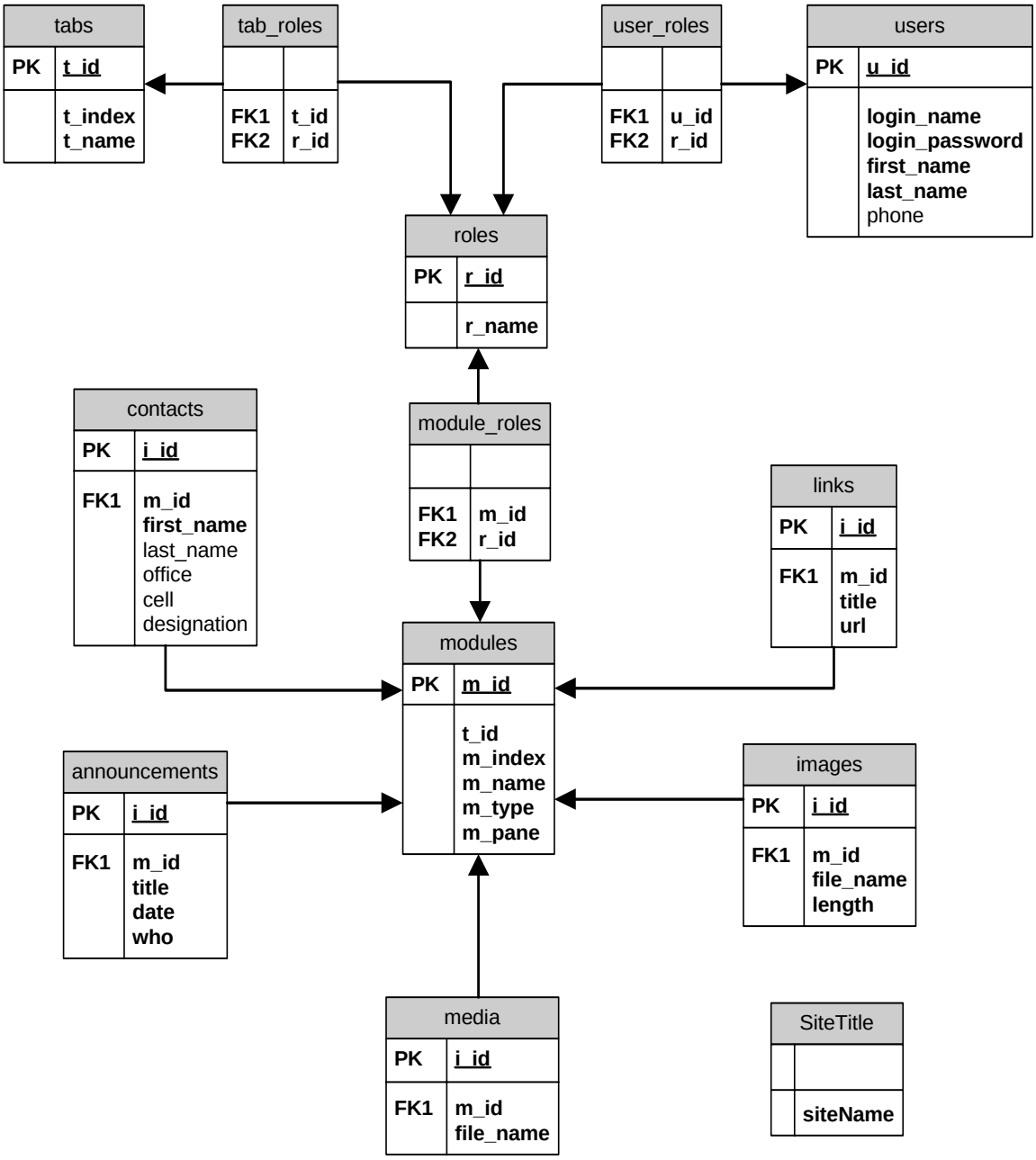
2.3 Framework

The entire portal framework has been built using JavaBeans and JSP. These along with a few servlets define all the modules. The core work of security and module placements on the page is also done using JavaBeans and servlets. The framework is such that new independent modules can be integrated into it without sacrificing uptime. This provides the extreme flexibility to create and organize a webpage. Intelligent modules can be created to display content in different formats depending on the whether the module is on the left or right pane in the page.

2.4 Database Schema

The portal database is organized in a way to provide role based security and hence the “Users” table along with a “Roles” table and “UserRoles” table become the most central aspect to the entire portal. It is around these tables that all other module data revolve i.e. each module’s table has a foreign key referencing the “Tabs” table to authenticate users. The “Users” table thus stores all vital portal user information along with the “Roles” table which stores all portal roles and “UserRoles” table which references both “Users” table and “UserRoles” table.

Content Management System
Database Schema



3.0 Managing Portal Security

The correct operation of the portal depends on managing security issues which are based on roles. By choosing role based security, we incorporate into the portal a work scenario where only groups of people are granted privileges to perform a particular operation. This enables multi-tier operation in places like banks where security is an issue. The portal authenticates user's identity and credentials against information stored in a database, and provides authorization, where a user's permissions for a requested resource are verified and granted. We shall analyze portal roles and its various features using a top-down approach. Figure 2 illustrates the top-down scenario of portal management.

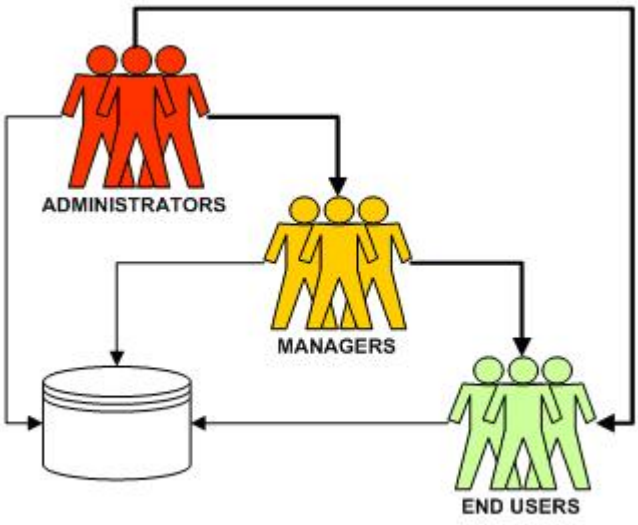


Figure 2: Managing security using portal hierarchy

3.1 Portal Administrators

The portal is managed by a group of one or more administrators with “Admin” roles who act as decision makers to assign roles to users. Administrators basically delegate the work and distribute it among module managers. They have the power to create new tabs on the fly and add new modules onto the tabs. Administrators can also create new roles while delegating work to those lower in the hierarchy. They constitute the top level of the portal hierarchy. Portal administration thus allows the user to perform a variety of site management and configuration tasks.

3.2 Module Managers

Module managers are those who have been assigned a role by the portal administrators to manage, edit and add content onto a module. They form the second tier of the portal architecture.

3.3 Portal Users

Portal users form the third and last tier of the three-tier hierarchy. They are assigned privileges/ roles by the portal administrators to only view tabs. All users to the portal have a unique identity to be authenticated against a database.

3.6 Access and Navigation Views

Access to the portal is based on authenticating unique email identities and their corresponding password. As can be seen, portal administrators have all the rights to view, edit, manage and add content to modules and, create new tabs.

Modules managers are authorized users with special permissions to edit a particular module in a particular tab. The administrator determines their access rights. As compared to module managers, a normal authorized user has only viewing privileges on tabs and no edit permissions. Portal administrators also determine a normal user's viewing privileges. Unauthorized users on the other hand can only view the home page of the portal. To obtain access they must first register and their registration request must be approved by the administrator who then provides necessary permissions. Figure 3 illustrates the various navigation views as seen by the administrator, authorized user and unauthorized user.

Public User's View	Authorized User's View	Administrator's View
Portal Menu Home Kidz Island Mickeyz Club Lion King Club Legends Family Fun	Portal Menu Home Kidz Island Mickeyz Club Lion King Club Legends Toon Shop Family Fun Movie Formula	Portal Menu Home Kidz Island Mickeyz Club Lion King Club Legends Movie Reviews Toon Shop Family Fun Movie Formula Admin Page

Figure 3: An instance of the portal navigation bar showing different views.

4.0 Client-Side Computations

The content management system has been designed such that the framework is dynamic and almost entirely database driven. As a result, the amount of client side computing that can be done is greatly reduced and every user click performs a round trip to the server and back. This increases the load on the server. Some client side computations come in handy as they avoid many unnecessary computations. The ones that have been used in the portal are discussed below.

4.1 JavaScript

JavaScript is used to interact with the HTML source code and dynamic content. They validate user form data before it is posted back to the server. Mostly, input format validation and data type checks are performed.

4.2 XML/ XSLT

XML/ XSLT has been used as an administrator only feature. A servlet is used to output the contents of the users table in the database. DOM API has been used to construct the XML document programmatically. XSLT is then used to format the XML to HTML and returned to the client browser.

4.3 Cascading Style Sheets

This file decides the styling features to apply when a page is rendered on the client browser. Style sheets are configured to recognize browsers and apply styles accordingly.

5.0 Portal Functionality

The various modules designed to perform different tasks form the essence of the portal's functionality. New features and functionality can be plugged in as the portal evolves. The table below describes some of the modules. While the "Default.jsp" forms the user interface for the content that is displayed, the "<ModuleName>.jsp" page forms the editable interface for that module as shown in figure 1.

MODULE NAME	DESCRIPTION
Contacts Module	Allows users to enter contacts details like name and phone numbers.
Images Module	Allows users to upload images onto the server. The uploaded file shows up as a thumbnail which when clicked, displays the entire image. This is an intelligent module which changes the way images are displayed according to whether the module is in the left or the right pane.
Links Module	Permits users to specify a URL to another site and give the link an appropriate name. This shows up a link with the user specified name on the UI.
Login Module	Allows the user to login to the portal.

Register Module	Allows users to register by filling a form. Registered users do not have any privileges until he is assigned one by the portal administrator.
Announcements Module	Allows users to post a group of announcements as a single module. Also shows the last modified date and the identity of the user who made the changes.

We now look at some of the administrative modules below.

MODULE NAME	DESCRIPTION
Users Module	It allows administrators to search by email and assign new roles to users. This module also allows administrators to get a list of registered portal users. This is an XML module that uses DOM API and XSLT for converting XML to HTML.
Roles Module	Permits the administrator to define new roles.
Tab Layout	This is the layout of the modules on the page seen by the administrator. New tabs can be created and their layouts are defined using this module.
Site Title	Allow administrators to rename the portal title. This immediately reflects in the standard header that is used in all the pages.

6.0 Performance Testing

Using jakarta-jmeter-2.0.3, the performance of the portal was analyzed to answer the following questions

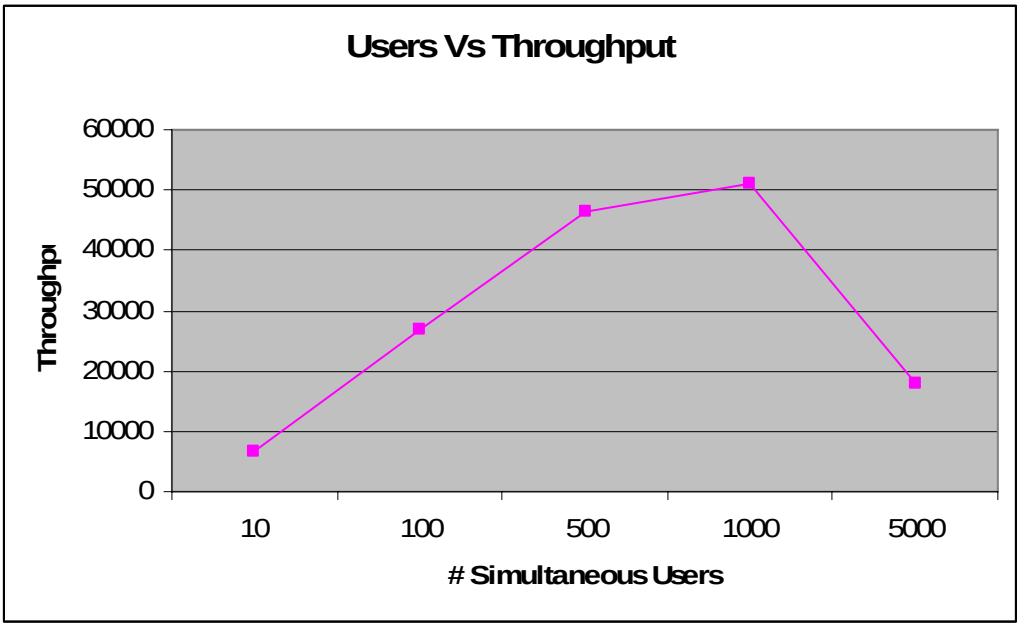
- What is the expected peak number of users the server can handle while providing a good response time.
- The maximum number of users the server can handle while still providing an acceptable response time.

6.1 Testing Scenario

Using jmeter, we analyzed the home page of the portal by creating a number of concurrent threads; starting after a ramp up period of 1 second, with each thread sending 10 requests one after the other. Here the threads are analogous to users. So, the first point plotted in the graph represents the number of requests handled by the server per minute, when 10 users log into the portal and send in 10 requests one after the other. We continued the analysis by increasing the number of users, while fixing the number of requests per user at 10. This is reasonable since each user could send utmost 10 requests to the server per minute and we are measuring the throughput of the server per minute.

6.2 Plot of the Jmeter Results

As explained above, the throughput/minute of the server was measured for 10, 100, 500, 1000, 5000 concurrent users. The results are shown in following graph.



6.3 Analysis of the Results

The curve has the shape of a throughput plot with the values on the y-axis increasing up to a point and decreasing at a steady state after that. The actual plot would have been a smooth elliptical curve if enough points where known. As we can see from the graph, at the beginning, the server can handle more number of incoming requests and the number of requests handled per minute also increases. This behavior reaches a peak limit at around 1000 users. Thereafter the server performance degenerates. Hence, the answers to our questions are as follows,

- The peak number of users the server can handle per minute while giving a good response time is 1000 users. This is evident from the peak point in the graph.
- The maximum number of users the server can handle per minute while giving an acceptable response time is around 4000 users. This is

because, from the plot we can see that the server can still handle these many users but the requests handled per minute is around 2500 which is half the value of the previous answer.

6.4 Some Relevant Issues

We have analyzed only the home page of the portal and tried to generalize it for the entire portal. This is reasonable since, the home page and all the other pages in the portal that generates content for the users are basically the same default.jsp page. Hence the results are valid across the entire portal.

6.5 Browser Issues

While developing the portal, Firefox was our browser of choice and midway through our development, we switched over to IE and found a number of errors. The source of these errors turned out to be the way the IE browser reacts to terminating characters coded using java. These errors have been fixed and the portal now fully supports both IE and Firefox.

7.0 Security Issues

The portal framework is such that there exists a single default.jsp page onto which modules are dynamically injected based on the query string in the URL. This allows for possible misuse and unauthorized access to data. Possible hackers can type in the URL along with a random query string value. Using a security bean, which checks the user identity stored in session variables and his permissions before continuing to fetch the page data, prevents this and redirects

the user to an error page. Similarly, the security bean also protects all the other administrator only pages from possible misuse.

7.1 Issues with Malicious Users

The problem arises because people with special privileges can add new modules and module content to individual tabs. There arises a situation where users can inject malicious code into the page. Such malicious code could be JavaScript and server script tags. These are discussed below.

- Injecting JavaScript code presents a real threat to the portal. We have found that such malicious injections can affect utmost one page. However, this affects all the users requesting that page.
- Injecting server script tags causes no problems as these are stored as static text in the database and are not interpreted when the page is being rendered.

The above problems are however isolated occurrences since the only users who have such privileges are administrators and managers who are closely associated with the portal. The users without any privileges do not pose any threat.

8.0 References

http://asp.net/Default.aspx?tabindex=8&tabid=47

http://dev.mysql.com/doc/

http://java.sun.com/reference/api/index.html

http://www.w3schools.com/