



Spring Framework

Training Assignments


Document Code	25e-BM/HR/HDCV/FSOFT
Version	1.1
Effective Date	20/11/2012

RECORD OF CHANGES

No	Effective Date	Change Description	Reason	Reviewer	Approver
1	01/Oct/2018	Add the new labs	Create new	DieuNT1	VinhNV
2	01/Jun/2019	Update template	Fsoft template	DieuNT1	VinhNV

Contents

Objectives	4
Business needs	4
Working Environments	4
Product Architecture	5
Technologies	5
Database Relationship	5
Assignment Descriptions	7

	<table><tr><td>CODE:</td><td>JSFW.L.A102 (Blogs)</td></tr><tr><td>TYPE:</td><td>LONG</td></tr><tr><td>LOC:</td><td>N/A</td></tr><tr><td>DURATION:</td><td>360 MINUTES</td></tr></table>	CODE:	JSFW.L.A102 (Blogs)	TYPE:	LONG	LOC:	N/A	DURATION:	360 MINUTES
CODE:	JSFW.L.A102 (Blogs)								
TYPE:	LONG								
LOC:	N/A								
DURATION:	360 MINUTES								

Objectives

After finishing the following exercises, trainees will:

- ✓ Understand Spring Core (IoC, DI)
- ✓ Practice with JDBC Templates to connect database in spring
- ✓ Use JPA for object-relational mapping
- ✓ Practice with the Spring MVC framework architecture
- ✓ Know how to write a Web application with Spring MVC Framework and Spring Data JPA

Business needs

Create a Web application based on HTML, CSS, JavaScript, Ajax, Servlet, JSP, Spring framework.

The blog system that we are going to develop is a single user system. The owner of the system will be able to perform the following actions:

- ✓ Login and logout
- ✓ Create, update and delete posts
- ✓ Published, unpublished and archive posts
- ✓ Approve and delete comments

All other users are guest users who can perform the following actions:

- ✓ Read posts
- ✓ Create comments

Additional Requirements for this system include:

- ✓ The homepage of the system should display a list of the most recent posts.
- ✓ If a page contains more than 10 posts, they should be displayed in pages.
- ✓ The system should display a post together with its comments.
- ✓ The system should be able to list posts with a specified tag.
- ✓ The system should show a cloud of tags indicating their use frequencies.
- ✓ The system should show a list of most recent comments.

Working Environments

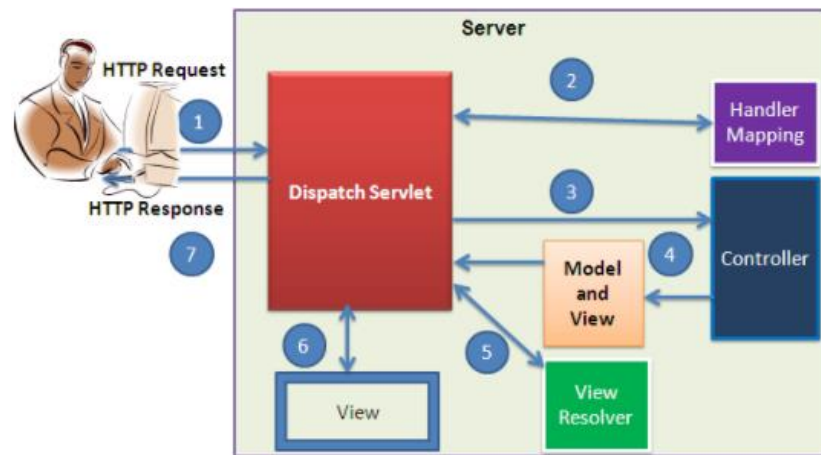
- ✓ JDK 1.8 +
- ✓ Maven 3+
- ✓ Eclipse IDE or STS,
- ✓ MySQL DB Server/MySQL,

- ✓ Apache TomCat 7 or later,
- ✓ Internet connection,

Delivery: Source code, deployment and testing, reviewing evidences packaged in a compress archive.

Product Architecture

Web applications are by nature distributed applications, meaning that they are programs that run on more than one computer and communicate through a network or server. Specifically, web applications are accessed with a web browser and are popular because of the ease of using the browser as a user client.



The product is implemented using MVC Pattern base on Spring Web MVC.

Technologies

The product implements one or more technology:

- ✓ HTML & CSS
- ✓ JavaScript & Ajax
- ✓ MVC & JSP Model
- ✓ Spring Framework: IoC, Bean, SpEL, JdbcTemplate, Spring Data JPA, Interceptor, Validation, Spring Security.

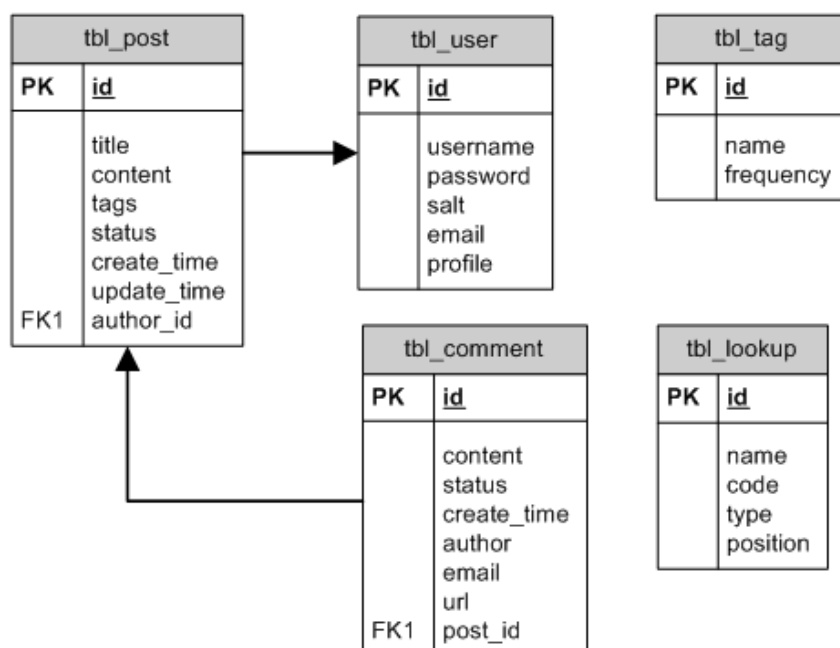
Database Relationship

Based on the analysis of the requirements, we decide to use the following database tables to store the persistent data for our blog application:

- ✓ **tbl_user** stores the user information, including username and password.
- ✓ **tbl_post** stores the blog post information. It mainly consists of the following columns:
 - title: required, title of the post;
 - content: required, body content of the post which uses the Markdown format;
 - status: required, status of the post, which can be one of following values:
 1. meaning the post is in draft and is not visible to public;
 2. meaning the post is published to public;

3. meaning the post is outdated and is not visible in the post list (still accessible individually, though).
- tags: optional, a list of comma-separated words categorizing the post.
- ✓ **tbl_comment** stores the post comment information. Each comment is associated with a post and mainly consists of the following columns:
- author: required, the author name;
 - email: required, the author email;
 - url: optional, the author website URL;
 - content: required, the comment content in plain text format.
 - status: required, status of the comment, which indicates whether the comment is approved (value 2) or not (value 1).
- ✓ **tbl_tag** stores post tag frequency information that is needed to implement the tag cloud feature. The table mainly contains the following columns:
- name: required, the unique tag name;
 - frequency: required, the number of times that the tag appears in posts.
- ✓ **tbl_lookup** stores generic lookup information. It is essentially a map between integer values and text strings. The former is the data representation in our code, while the latter is the corresponding presentation to end users. For example, we use integer 1 to represent the draft post status and stringDraft to display this status to end users. This table mainly contains the following columns:
- name: the textual representation of the data item that is to be displayed to end users;
 - code: the integer representation of the data item;
 - type: the type of the data item;
 - position: the relative display order of the data item among other items of the same type.

The following entity-relation (ER) diagram shows the table structure and relationships about the above tables



Assignment Descriptions

1. Create the pages as below:

- ✓ Home page:

The screenshot shows the Home page of a blog. At the top is a navigation bar with links: Home, About, Contact, and Login. The main content area features a 'Welcome!' message, a post by 'demo' from January 2, 2009, and a 'A Test Post' by 'demo' from the same date. Both posts include a description, tags, and a 'Comments' link. A sidebar on the right contains 'Tags' (blog, test) and 'Recent Comments' (Tester on A Test Post). The footer contains the copyright notice: 'Copyright © 2016 by My Company. All Rights Reserved.'

- ✓ Login page:

The screenshot shows the Login page of a blog. At the top is a navigation bar with links: Home, About, Contact, and Login. The main content area features a 'Login' heading, a form with fields for 'Username' (demo) and 'Password' (masked with dots), a 'Remember me next time' checkbox, and a 'Login' button. The footer contains the copyright notice: 'Copyright © 2016 by My Company. All Rights Reserved.'

- ✓ About page

The screenshot shows the About page of a blog. At the top is a navigation bar with links: Home, About, Contact, and Login. The main content area features an 'About' heading and a paragraph: 'This is the "about" page for my blog site.' The footer contains the copyright notice: 'Copyright © 2016 by My Company. All Rights Reserved.'

✓ Contact Page

Blog

[Home](#) [About](#) [Contact](#) [Login](#)

[Home](#) » [Contact](#)

Contact Us

If you have business inquiries or other questions, please fill out the following form to contact us. Thank you.

*Fields with * are required.*

Name *

Email *

Subject *

Body *

✓ Create post

Blog

[Home](#) [About](#) [Contact](#) [Logout \(demo\)](#)

[Home](#) » [Create Post](#)

Create Post

*Fields with * are required.*

Title *

Content *

You may use [Markdown syntax](#).

Tags

Please separate different tags with commas.

Status *

Draft

demo

[Create New Post](#)
[Manage Posts](#)
[Approve Comments \(0\)](#)
[Logout](#)

Tags

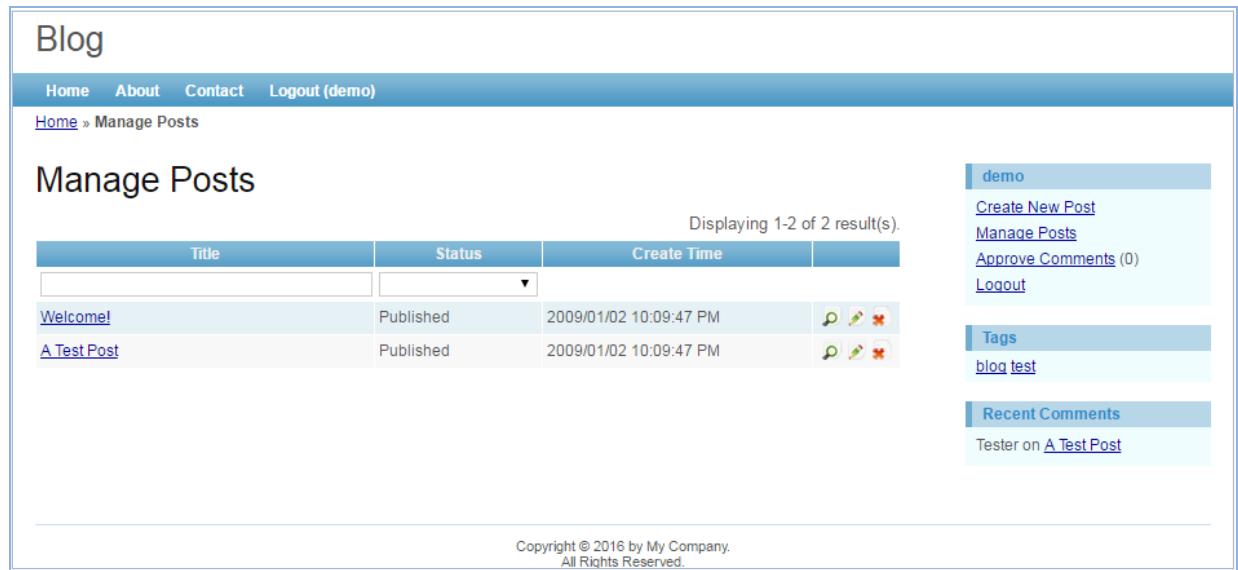
[blog test](#)

Recent Comments

Tester on [A Test Post](#)

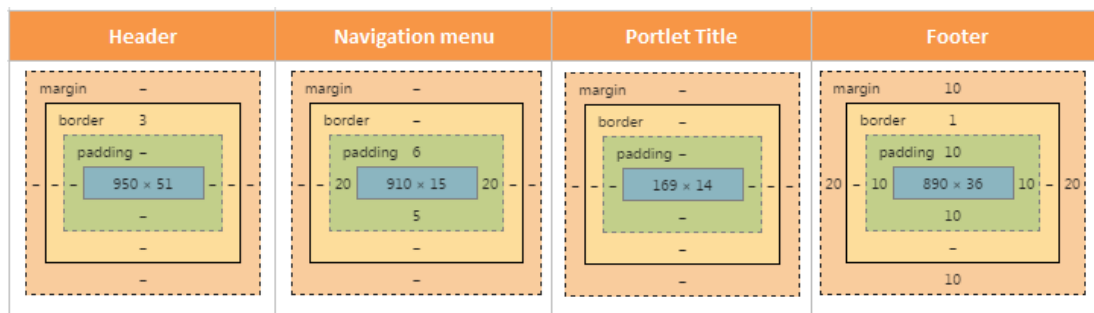
Copyright © 2016 by My Company.
All Rights Reserved.

- ✓ Manage post page



2. Screen Requirement:

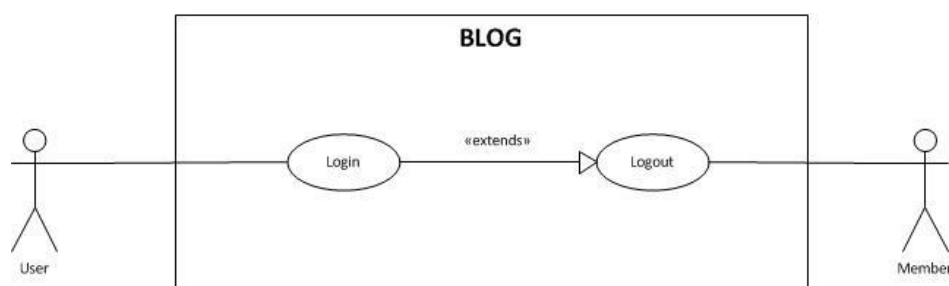
CSS Box Model as below:



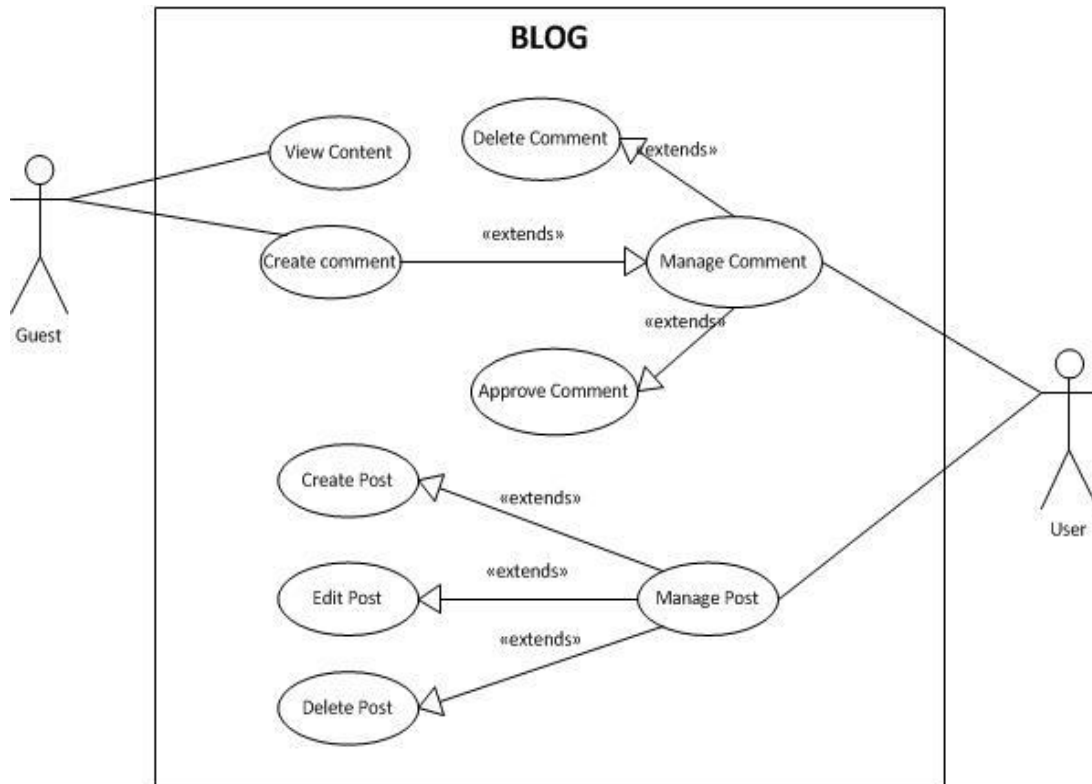
- ✓ Let design based The screen illustrated.
- ✓ Add User must be a popup.
- ✓ Placeholder text in inputs has a light gray color. You need to define the placeholder for input items on the above screen.
- ✓ Add a new file fStyle.css to customize your screen, it will based on convention of CSS style.
- ✓ The **Index** page must contain 3 parts: Banner, Menu, Body

3. Functional Requirement:

As you may have found that the skeleton application already provides user authentication by checking if the username and password are both **demo** or **admin**. In this section, we will modify the corresponding code so that the authentication is done against the User database table.



In our blog application, a post may be displayed among a list of posts or by itself. Managing posts mainly refers to listing posts in an administrative view that allows us to see posts with all statuses, updating them and deleting them. In this assignment, you must finish the feature in use case as below:



-- THE END --