

## **Project Summary**

A Blog web system with SpringMVC + Maven + Hibernate + MySQL with Annotation mapping. In this system, users can create own account, login the system. They can also create own passage, update or edit their passage, delete their passage. Moreover, users can view both own passages and passages wrote by others. User can comment on those passage and press 'like' button to like the passage.

## **Summary of Functionality Performed**

A User Can:

1. Register
2. Login: Use name+password
3. Create own passage: with title+description+content
4. View own passage
5. Edit own passage
6. Delete own passage
7. View all passages in blog : Read Only
8. Comment on passages
9. 'Like' one passage : 1 user for 1 chance to thumb

## **Summary of the technologies used**

Spring MVC + Maven + AJAX + Annoation Mapping + MySQL + Hibernate

1. Use spring form to inject a java bean entity
2. Use filter to check user session
3. Use validator to validate the null value in controller
4. Use Annotation mapping in controller implement hyperlink and jump between jsp pages
5. Use Hibernate Query, criteria in DAO class to implement query data from mySQL database
6. Use AJAX in validate duplicate user name (user name already exists in database) when user register.
7. Configure Maven environment

## Roles in this system

User:

User	Passage	Comment
<ul style="list-style-type: none"><li>userId BIGINT(20)</li><li>email VARCHAR(255)</li><li>name VARCHAR(255)</li><li>password VARCHAR(255)</li><li>phoneNum INT(11)</li></ul>	<ul style="list-style-type: none"><li>passageId BIGINT(20)</li><li>content VARCHAR(255)</li><li>discription VARCHAR(255)</li><li>thumbNum INT(11)</li><li>title VARCHAR(255)</li><li>USER_ID BIGINT(20)</li></ul>	<ul style="list-style-type: none"><li>commentId BIGINT(20)</li><li>content VARCHAR(255)</li><li>PASSAGE_ID BIGINT(20)</li><li>USER_ID BIGINT(20)</li></ul>
<b>Indexes</b>	<b>Indexes</b>	<b>Indexes</b>
PRIMARY	PRIMARY FKb9h48xlmr7nhmigsitentss8r	PRIMARY FKanvlftgtgwvayvxgax6wmp5nr FKqq94fmcnnt9e6adcj05p90bhv

@OneToMany → user – passage / passage -- comment

@ManyToMany → userLikes(the Passages user likes) – Passages liked by user

Passage_User_Like
<ul style="list-style-type: none"><li>userLikes_passageId BIGINT(20)</li><li>likedByList_userId BIGINT(20)</li></ul>
<b>Indexes</b>
PRIMARY FK33bpgpi7mwi1f0ahqmp3ummau

(This is a joint table)

## Screen shots

Login|Register

User Name:

Password:

User Login

---

[Login](#)|[Register](#)

User Name:

Password:

Email:

Phone number:

User register (duplicated/ null value validator ready!)

---

[View-My-Passage](#)

Leyi

Leyi2

Leyi3

You can view all passage here (read + comment +like only )

---

Title: "Leyi"

Description: "1"

Content: "1"

User: "LeyiWang" Says: "I am Real Leyi"

[Back To View](#)

---

Title: "Leyi"

Description: "1"

Content: "1"

User: "Leyi" Says: "No I am Real!!!!"

User: "LeyiWang" Says: "I am Real Leyi"

[Back To View](#)

You can comment and others comments also shows here !

## View-Public-Passage

"Leyi" >

"Leyi2" >

"Leyi3" >

In own space, you can create edit and delete passage you own.

## Code part:

### 1. com.me.controller HomeController

```
package com.me.controller;
```

```
import java.util.Locale;
```

```
import org.slf4j.Logger;
```

```
import org.slf4j.LoggerFactory;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestMethod;
```

```
/**
```

```
 * Handles requests for the application home page.
```

```
 */
```

```
@Controller
```

```
@RequestMapping(value = "/")
```

```
public class HomeController {
```

```
    private static final Logger logger = LoggerFactory.getLogger(HomeController.class);
```

```
    /**
```

```
     * Simply selects the home view to render by returning its name.
```

```
     */
```

```
    @RequestMapping(value = "/", method = RequestMethod.GET)
```

```

        public String home(Locale locale, Model model) {
            logger.info("Welcome home! The client locale is {}.", locale);

            return "home";
        }
    }
}

```

## PassageController

```

package com.me.controller;

import java.util.ArrayList;
import java.util.List;
import java.util.Set;

import javax.annotation.Resource;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.validation.annotation.Validated;
import org.springframework.web.bind.WebDataBinder;
import org.springframework.web.bind.annotation.InitBinder;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

import com.me.dao.PassageDao;
import com.me.dao.UserDao;
import com.me.exception.CommentException;
import com.me.exception.PassageException;
import com.me.exception.UserException;
import com.me.pojo.Comment;
import com.me.pojo.Passage;
import com.me.pojo.User;
import com.me.validator.PassageValidator;
import com.me.validator.UserValidator;

```

```

@Controller
@RequestMapping("/user")
public class PassageController {
    @Autowired
    PassageValidator passagevalidator;

    @Autowired
    PassageDao passageDao;

    // Set a form validator
    @InitBinder
    protected void initBinder(WebDataBinder binder) {
        binder.setValidator(passagevalidator);
    }

    @RequestMapping(value="createPassage.htm", method = RequestMethod.GET)
    public String formView(ModelMap model,  HttpServletRequest req) {

        Passage passage = new Passage();
        model.addAttribute("passage", passage);
        return "createPassage";
    }

    @RequestMapping(value="deletePassage.htm", method = RequestMethod.GET)
    public ModelAndView deleteformView(ModelMap model,  HttpServletRequest req)
    throws PassageException {
        HttpSession session = req.getSession();
        User user = (User)session.getAttribute("user");
        if(user == null) {
            return new ModelAndView("login");
        }else {
            long id =Long.parseLong(req.getParameter("passageId"));
            System.out.println(user);
            ArrayList<Passage> passages = (ArrayList<Passage>)
passageDao.deletePassage(id,user);
            return new ModelAndView("viewOwnPassage","passageList",passages);
        }
    }

    @RequestMapping(value="viewAllPassage.htm", method = RequestMethod.GET)
    public ModelAndView viewAllPassageView(ModelMap model, Passage passage,
    HttpServletRequest req) throws PassageException {
        ArrayList<Passage> passages = (ArrayList<Passage>) passageDao.getPassages();
    }
}

```

```

        req.setAttribute("passageList", passages);
        return new ModelAndView("viewAllPassage","passageList",passages);
    }

```

//在这里加👍

```

    @RequestMapping(value="viewAllPassage.htm", method = RequestMethod.POST)
    public ModelAndView viewAllPassageLikeView(ModelMap model, Passage passage,
        HttpServletRequest req) throws PassageException, UserException {
        HttpSession s = req.getSession();
        if(s.getAttribute("user")!=null) {
            User user=(User)s.getAttribute("user");
            long passageId = Long.parseLong(req.getParameter("passageId"));
            passage = passageDao.get(passageId);
            int i = passageDao.likeArticle(passage, user);
            ArrayList<Passage> passages = (ArrayList<Passage>)
passageDao.getPassages();
            req.setAttribute("passageList", passages);
            return new ModelAndView("viewAllPassage","passageList",passages);
        }
        return new ModelAndView("login");
    }

```

```

    @RequestMapping(value="viewOwnPassage.htm", method = RequestMethod.GET)
    public ModelAndView viewOwnPassageView(ModelMap model, Passage passage,
        HttpServletRequest req) throws PassageException {
        HttpSession session = req.getSession();
        User user = (User)session.getAttribute("user");
        if(user == null) {
            return new ModelAndView("login");
        }else {
            ArrayList<Passage> passages = (ArrayList<Passage>)
passageDao.getOwnPassages(user);
            req.setAttribute("passageList", passages);
            return new ModelAndView("viewOwnPassage","passageList",passages);
        }
    }
}

```

```

    @RequestMapping(value="editOwnPassage.htm", method = RequestMethod.POST)
    public String updateFormView(ModelMap model, Passage passage, HttpServletRequest
req) throws PassageException {
        HttpSession s = req.getSession();
        if(s.getAttribute("user")!=null) {
            User user=(User)s.getAttribute("user");

```

```

        passage.setUser(user);
        long passageld = Long.parseLong(req.getParameter("passageld"));
        passage = passageDao.get(passageld);
        model.addAttribute("passage", passage);
        s.setAttribute("passage", passage);

        return "editOwnPassage";
    }
    return "login";
}

```

```

@RequestMapping(value="viewOnePublicPassage.htm", method =
RequestMethod.GET)
public ModelAndView viewPublicPassageCommentFormView(ModelMap model,
Passage passage, HttpServletRequest req) throws PassageException, CommentException {
    HttpSession s = req.getSession();
    if(s.getAttribute("user")!=null) {
        String commentContent = req.getParameter("commentContent");
        if(commentContent!=null) {
            User user=(User)s.getAttribute("user");
            long passageld = Long.parseLong(req.getParameter("passageld"));
            passage = passageDao.get(passageld);
            Set<Comment> commentList = passageDao.addComment(passage,
commentContent, user);
            s.setAttribute("passage", passage);
            return new
ModelAndView("viewOnePublicPassage","commentList",commentList);
        }
        return new ModelAndView("viewAllPassage");
    }
    return new ModelAndView("login");
}

```

```

@RequestMapping(value="viewOnePublicPassage.htm", method =
RequestMethod.POST)
public String viewPublicPassageFormView(ModelMap model, Passage passage,
HttpServletRequest req) throws PassageException {
    HttpSession s = req.getSession();
    if(s.getAttribute("user")!=null) {
        User user=(User)s.getAttribute("user");
        passage.setUser(user);
    }
}

```



```

        long passageld = Long.parseLong(req.getParameter("passageld"));
        passage = passageDao.get(passageld);
        model.addAttribute("passage", passage);
        s.setAttribute("passage", passage);

        return "viewOnePublicPassage";
    }
    return "login";
}

// @ModelAttribute("passage") Passage passage,

@RequestMapping(value="create-passage-success.htm", method =
RequestMethod.POST)
public ModelAndView successCreateView(@Validated @ModelAttribute("passage")
Passage passage, BindingResult bindingResult, ModelMap model,HttpServletRequest req)
throws UserException, PassageException {
    //如果用户名或密码不符合 返回 create Passage
    if (bindingResult.hasErrors()) {
        return new ModelAndView("createPassage"); //the are validation errors, go to
the form view
    }
    HttpSession s = req.getSession();
    if(s.getAttribute("user")!=null) {
        User user=(User)s.getAttribute("user");
        passage.setUser(user);
        Passage pass = passageDao.createPassage(passage);
        ArrayList<Passage> passages = (ArrayList<Passage>)
passageDao.getOwnPassages(user);
        req.setAttribute("passageList", passages);
        return new ModelAndView("viewOwnPassage","passageList", passages);
    }

// session user 不存在 去登陆
    return new ModelAndView("login");
}

@RequestMapping(value="viewOwnPassage.htm", method = RequestMethod.POST)
public ModelAndView successUpdateView(@Validated @ModelAttribute("passage")
Passage passage, BindingResult bindingResult, ModelMap model,HttpServletRequest req)
throws UserException, PassageException {
    //如果用户名或密码不符合 返回自己
    if (bindingResult.hasErrors()) {
        return new ModelAndView("editOwnPassage"); //the are validation errors, go

```

to the form view

```
    }
    long passageld = Long.parseLong(req.getParameter("pld"));
    Passage p = passageDao.get(passageld);
    HttpSession session = req.getSession();
    User user = (User)session.getAttribute("user");
    String content = passage.getContent();
    p.setContent(content);
//    (Passage)session.getAttribute("passage")
    if(user == null) {
        return new ModelAndView("login");
    }else {
        if(passageDao.updatePassage(p)) {
            ArrayList<Passage> passages = (ArrayList<Passage>)
passageDao.getOwnPassages(user);
            req.setAttribute("passageList", passages);
            System.out.print(passage.getContent());

            return new ModelAndView("viewOwnPassage", "passageList", passages);
        }else {
            return new ModelAndView("login");
        }
    }
}

}
```

## Login controller

```
package com.me.controller;

import java.util.ArrayList;
import java.util.List;

import javax.annotation.Resource;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.validation.annotation.Validated;
import org.springframework.web.bind.WebDataBinder;
import org.springframework.web.bind.annotation.InitBinder;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import com.me.dao.PassageDao;
import com.me.dao.UserDao;
import com.me.exception.PassageException;
import com.me.exception.UserException;
import com.me.pojo.Passage;
import com.me.pojo.User;
import com.me.validator.UserValidator;
```

@Controller

```
public class UserLoginController {
    @Autowired
    UserValidator uservalidator;

    @Autowired
    UserDao userDao;

    @Autowired
    PassageDao passageDao;

    // Set a form validator
    @InitBinder
    protected void initBinder(WebDataBinder binder) {
        binder.setValidator(uservalidator);
    }

    @RequestMapping(value="login.htm", method = RequestMethod.GET)
    public String formView(ModelMap model, User user, HttpServletRequest req) {
        model.addAttribute("user", user);
        return "login";
    }

    @RequestMapping(value="user.htm", method = RequestMethod.POST)
    public String successView(@Validated @ModelAttribute("user") User user, BindingResult
bindingResult, ModelMap model,HttpServletRequest req) throws UserException,
```

```

PassException {

    //如果用户名或密码不符合 返回 "login"
    if (userDao.login(user.getName(), user.getPassword())==null ) {          // 差 个
validation !!!!

        return "login"; //the are validation errors, go to the form view
    }
    user = userDao.login(user.getName(), user.getPassword());
    //no errors, so go to the success view
    HttpSession s = req.getSession();
    s.setAttribute("user", user);
    Passage passage = new Passage();
//    model.addAttribute("passage",passage);
    req.setAttribute("passage", passage);
    ArrayList<Passage> passages = (ArrayList<Passage>) passageDao.getPassages();
    req.setAttribute("passageList", passages);
    return "viewAllPassage";
}

}

```

## RegisterController

```

package com.me.controller;

import java.io.IOException;

import javax.annotation.Resource;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.validation.annotation.Validated;
import org.springframework.web.bind.WebDataBinder;
import org.springframework.web.bind.annotation.InitBinder;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;

```

```

import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

import com.me.dao.UserDao;
import com.me.exception.UserException;
import com.me.pojo.User;
import com.me.validator.UserValidator;

@Controller
@RequestMapping("/register.htm")
public class UserRegisterController {
    @Autowired
    UserValidator uservalidator;

    @Autowired
    UserDao userDao;

    // Set a form validator
    @InitBinder
    protected void initBinder(WebDataBinder binder) {
        binder.setValidator(uservalidator);
    }

    @RequestMapping( method = RequestMethod.GET)
    // @RequestMapping("/register.htm")
    public String formView(ModelMap model, User user, HttpServletRequest req) {

        model.addAttribute("user", user);
        return "register";
    }

    @RequestMapping(method = RequestMethod.POST)
    public String successView(@Validated @ModelAttribute("user") User user, BindingResult
bindingResult, ModelMap model,HttpServletRequest req ,HttpServletResponse resp) throws
UserException, IOException {
        if (bindingResult.hasErrors()) {
            return "register"; //the are validation errors, go to the form view
        }
        if(userDao.isUserExsist(req.getParameter("name"))){
            req.setAttribute("alert", "Dupulicate!!");
            return "register";
        }
        //no errors, so go to the success view

```

```

        req.setAttribute("alert", "");
        userDao.register(user);
        return "user-success";
    }
}

```

## View

### createPassage.jsp

```

<%@ page session="true" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <!-- Bootstrap CSS -->
    <link
                                                                    rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <meta charset="UTF-8">
    <title>Login</title>
</head>
<body>
<form:form modelAttribute="passage" action="/userapp/user/create-passage-success.htm"
method="POST">
<p><a
            href="createPassage.htm">Create
                A
                Passage</a>|<a
href="home.htm">Update(meixiehao)</a></p><br/>
<div id="createP">
Passage Title: <form:input path="title"/> <form:errors path="title"/><br/>
Description: <form:input path="discription"/> <form:errors path="discription"/><br/>
Content: <form:input path="content"/> <br/>

```

```
<input type="submit" value="create"/>
</div>
</form:form>
```

```
</body>
</html>
```

### **editownPassage.jsp**

```
<%@ page session="true" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
<head>
<!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <!-- Bootstrap CSS -->
    <link                                     rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
<script>
function myFunction(){
    if(window.confirm("You Sure Want delete this?")){
        window.location.href                                     =
"http://localhost:8080/userapp/user/deletePassage.htm?passageId=${passage.passageId}";
    }else{
        window.location.href = "http://www.baidu.com";
    }
}
</script>
<meta charset="UTF-8">
<title>EDIT</title>
</head>
<body>

<form:form                                     modelAttribute="passage"
action="/userapp/user/viewOwnPassage.htm?pld=${passage.passageId}" method="POST">
    Passage Title: <form:input path="title" value="${passage.title }"/> <form:errors
path="title"/><br/>
```

```

        Description: <form:input path="discription" value="${passage.discription }"/>
<form:errors path="discription"/><br/>
        Content: <form:input path="content" value="${passage.content }"/> <br/>
        <input type="submit" value="update"/>
        <button type="button" onclick="myFunction()" >Delete</button>
</form:form>

```

```

</body>
</html>

```

## Home

```

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ page session="false" %>
<html>
<head>
<!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <!-- Bootstrap CSS -->
    <link
                                                                    rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <title>Home</title>
</head>
<body>
<a href="login.htm">Click Here to go to the USER FORM VIEW</a>
</body>
</html>

```

## Login.jsp

```

<%@ page session="true" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>

```



```

<head>
<!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <!-- Bootstrap CSS -->
    <link
                                                                    rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">

<meta charset="UTF-8">
<title>Login</title>
</head>
<body>
<form:form commandName="user" action="user.htm" method="POST">
<p><a href="login.htm">Login</a>|<a href="register.htm">Register</a></p><br/>
<div id="login">
User Name: <form:input path="name"/> <form:errors path="name"/><br/>
Password: <form:input path="password"/> <form:errors path="password"/><br/>
<input type="submit" value="login"/>
</div>
</form:form>

</body>
</html>

```

## Register.jsp

```

<%@ page session="true" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

```

```

<!-- Bootstrap CSS -->
<link
rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
<title>Register</title>
<meta charset="utf-8">
<script>
function loadXMLDoc()
{
    var xmlhttp;
    if (window.XMLHttpRequest)
    {
        // IE7+, Firefox, Chrome, Opera, Safari 浏览器执行代码
        xmlhttp=new XMLHttpRequest();
    }
    else
    {
        // IE6, IE5 浏览器执行代码
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange=function()
    {
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {
            document.getElementById("register").innerHTML=xmlhttp.responseText;
        }
    }
    xmlhttp.open("POST","register.htm",true);
    xmlhttp.send();
}
</script>
</head>

```

```

<body>
<div id="register">
<form:form commandName="user" action="register.htm" method="POST">
<p><a href="login.htm">Login</a>|<a href="register.htm">Register</a></p><br/>
User   Name:   <form:input   name="name"   path="name"   />   <form:errors
path="name"/><label>"${requestScope.alert}"</label><br/>
Password:   <form:input   type="password"   path="password"/>   <form:errors

```

```

path="password"/><br/>
Email: <form:input type="email" path="email"/> <form:errors path="email"/><br/>
Phone number: <form:input path="phoneNum"/> <form:errors path="phoneNum"/><br/>
<button onclick="loadXMLDoc()">AJAX Button</button>
</form:form>
</div>
</body>

</body>
</html>

```

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

<!-- Bootstrap CSS -->
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
</head>
<body>
Welcome ${requestScope.user.name} ${requestScope.user.email}
${requestScope.user.password} ${requestScope.user.phoneNum}
</body>
</html>

```

## viewAllPages.jsp

```

<%@ page session="true" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<%@ taglib prefix = "c" uri = "http://java.sun.com/jsp/jstl/core" %>

```

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Login</title>
<!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
</head>
<body>
<a href="/userapp/user/viewOwnPassage.htm">View-My-Passage</a>
<!-- 只能看点赞评论 不能查/修改的公共区域 -->
<c:forEach var="passage" items="${requestScope.passageList}">
    <form action="/userapp/user/viewOnePublicPassage.htm" method="POST" >
        <label >${passage.title}</label>
        <input type="hidden" name="passageId" value="${passage.passageId}">
        <input type="submit" value="view"/>
    </form>
    <form action="/userapp/user/viewAllPassage.htm" method="POST" >
        <input type="submit" value="${passage.thumbNum }"></br>
        <input type="hidden" name="passageId" value="${passage.passageId}">
    </form>
</c:forEach>
</body>
</html>

```

### viewOwnPublicPassage

```

<%@ page session="true" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<%@ taglib prefix = "c" uri = "http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Login</title>
<!-- Required meta tags -->

```

```

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <!-- Bootstrap CSS -->
    <link
                                                                    rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
</head>
<body>
<a href="/userapp/user/viewOwnPassage.htm">View-My-Passage</a>
<!-- 只能看点赞评论 不能查/修改的公共区域 -->
<c:forEach var="passage" items="${requestScope.passageList}">
    <form action="/userapp/user/viewOnePublicPassage.htm" method="POST" >
        <label >${passage.title}</label>
        <input type="hidden" name="passageId" value="${passage.passageId}">
        <input type="submit" value="view"/>
    </form>
    <form action="/userapp/user/viewAllPassage.htm" method="POST" >
        <input type="submit" value="${passage.thumbNum }"></br>
        <input type="hidden" name="passageId" value="${passage.passageId}">
    </form>
</c:forEach>
</body>
</html>

```

### viewOnePassage.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix = "c" uri = "http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>View Public Passage</title>
<!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <!-- Bootstrap CSS -->

```

```

        <link                                                    rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
</head>
<body>
    <label>Title: "${sessionScope.passage.title}"</label><br/>
    <label>Description: "${sessionScope.passage.description}"</label><br/>
    <label>Content: "${sessionScope.passage.content}"</label><br/>
    <c:forEach var="comment" items="${requestScope.commentList}">
        <label>User: "${comment.user }"</label>
        <label>  Says: "${comment.content }"</label><br/>
    </c:forEach>

    <form action="/userapp/user/viewOnePublicPassage.htm" method="get">
        <input type="text" name="commentContent">
        <input type="hidden" name = "passageId" value="${passage.passageId}">
        <input type="submit" value="comment">
    </form>
    <a href="viewAllPassage.htm">Back To View</a>

</body>
</html>

```

### ViewOwnPassage.jsp

```

<%@ page session="true" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<%@ taglib prefix = "c" uri = "http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
<head>
<!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
<script>
function myFunction(){
    window.location.href = "http://localhost:8080/userapp/user/createPassage.htm"
}

```

```

</script>
<meta charset="UTF-8">
<title>Login</title>
</head>
<body>
<a href="/userapp/user/viewAllPassage.htm">View-Public-Passage</a>
<!-- user 自己写的 能看能查能改 -->
<div id="viewOwnPassage">
    <c:forEach var="passage" items="${requestScope.passageList}">
        <form method="POST" action="/userapp/user/editOwnPassage.htm">
            <label>"${passage.title}" ></label>
            <input type="hidden" name="passageId" value="${passage.passageId}" />
            <input type="submit" value="Edit" >
            <br/>
        </form>
    </c:forEach>
    <br/>
    <button type="button" onclick="myFunction()" >Create</button>
</div>

</body>
</html>

```

## DAO

```

package com.me.dao;

import org.hibernate.HibernateException;

import com.me.exception.CommentException;
import com.me.pojo.Comment;

public class CommentDao extends DAO {
    //Create a comment
    public Comment createPassage(Comment comment) throws CommentException {
        try {
            begin();
            getSession().merge(comment);
            commit();
            return comment;
        } catch (HibernateException e) {
            rollback();
            throw new CommentException("Exception while creating user: " +

```

```

e.getMessage());
    }
}

}

```

```

package com.me.dao;

```

```

import java.util.logging.Level;
import java.util.logging.Logger;

```

```

import org.hibernate.HibernateException;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

```

```

public class DAO {

```

```

    private static final Logger log = Logger.getAnonymousLogger();

```

```

    private static final ThreadLocal sessionThread = new ThreadLocal();
    private static final SessionFactory sessionFactory = new
Configuration().configure("hibernate.cfg.xml").buildSessionFactory();

```

```

    protected DAO() {
    }

```

```

    public static Session getSession()
    {

```

```

        Session session = (Session) DAO.sessionThread.get();

```

```

        if (session == null)
        {
            session = sessionFactory.openSession();
            DAO.sessionThread.set(session);
        }
        return session;
    }

```

```

    protected void begin() {

```



```

        getSession().beginTransaction();
    }

    protected void commit() {
        getSession().getTransaction().commit();
    }

    protected void rollback() {
        try {
            getSession().getTransaction().rollback();
        } catch (HibernateException e) {
            log.log(Level.WARNING, "Cannot rollback", e);
        }
        try {
            getSession().close();
        } catch (HibernateException e) {
            log.log(Level.WARNING, "Cannot close", e);
        }
        DAO.sessionThread.set(null);
    }

    public static void close() {
        getSession().close();
        DAO.sessionThread.set(null);
    }
}

```

```
package com.me.dao;
```

```
import java.util.List;
import java.util.Set;
```

```
import org.hibernate.HibernateException;
import org.hibernate.Query;
```

```
import com.me.exception.CommentException;
import com.me.exception.PassageException;
import com.me.exception.UserException;
import com.me.pojo.Comment;
import com.me.pojo.Passage;
import com.me.pojo.User;
```

```
public class PassageDao extends DAO {
```

```

//create a passage
public Passage createPassage(Passage passage) throws PassageException {
    try {
        begin();
        getSession().merge(passage);
        commit();
        return passage;
    } catch (HibernateException e) {
        rollback();
        throw new PassageException("Exception while creating user: " +
e.getMessage());
    }
}

//delete a passage
public List<Passage> deletePassage(long passageld,User user) throws
PassageException{

    Passage passage = get(passageld);
    passage.setUser(null);
    Set<Passage> passageList = user.getPassageCatelog();
    Set<Passage> userlikepassageList = user.getUserLikes();
    passageList.remove(passage);
    userlikepassageList.remove(passage);
    user.setPassageCatelog(passageList);
    user.setUserLikes(userlikepassageList);
    try {
        begin();

//        commit();
//        begin();
        getSession().delete(passage);
        commit();

        List<Passage> passages = getOwnPassages(user);
        return passages;
    } catch (HibernateException e) {
        rollback();
        throw new PassageException("Exception while deleting user: " +
e.getMessage());
    }
}

```

```

//Update a passage
public boolean updatePassage(Passage passage) throws PassageException {
    try {
        passage = get(passage.getPassageId());
        begin();
        getSession().update(passage);
        commit();
        return true;
    } catch (HibernateException e) {
        rollback();
        throw new PassageException("Exception while updating user: " +
e.getMessage());
    }
}

```

```

//View a passage
public List<Passage> getPassages() throws PassageException{
    List<Passage> passages;
    try{
        begin();
        String hql = "from Passage";
        Query query = getSession().createQuery(hql);
        query.setMaxResults(10);
        passages = query.list();
        commit();
//        getSession().flush();
    }catch(Exception e){
        super.rollback();
        throw new PassageException("Get all public Passage dao error", e.getCause());
    }
    return passages;
}

```

```

//Get Own Passage List
public List<Passage> getOwnPassages(User user) throws PassageException{
    List<Passage> passages;

    try{
        begin();
        String hql = "from Passage where USER_ID = :userId";
        Query query = getSession().createQuery(hql);
        query.setLong("userId", user.getUserId());
        query.setMaxResults(10);

```

```

        passages = query.list();
        commit();
//        getSession().flush();
    }catch(Exception e){
        super.rollback();
        throw new PassageException("get all private Passage dao error", e.getCause());
    }
    return passages;
}

```

```

//Get Passage in passageld
public Passage get(long passageld) throws PassageException {
    try {
        begin();
        Query q = getSession().createQuery("from Passage where passageld
= :passageld");
        q.setLong("passageld", passageld);
        Passage passage = (Passage) q.uniqueResult();
        commit();
        return passage;
    } catch (HibernateException e) {
        rollback();
        throw new PassageException("Could not get Passage " + passageld, e);
    }
}

```

```

//add comment
public Set<Comment> addComment(Passage passage, String comment, User user)
throws CommentException{
    Set<Comment> commentList;
    try{
        begin();
        commentList = passage.getCommentCatalog();
        Comment comments = new Comment();
        comments.setContent(comment);
        comments.setPassage(passage);
        comments.setUser(user);
        commentList.add(comments);
        getSession().save(comments);
        commit();
//        getSession().flush();
    }catch(Exception e){
        super.rollback();
        throw new CommentException("Passage dao error in add comment",

```

```

        e.getCause());
    }
    return commentList;
}

//liked
public int likeArticle(Passage passage, User user) throws UserException{
    int likeNum;
    String errorMessage=null;
    Set<User> likedBy = passage.getLikedByList();
    try{
        super.begin();
        likedBy.add(user);
        passage.setLikedByList(likedBy);
        likeNum = passage.getLikedByList().size();
        passage.setThumbNum(likeNum);
        getSession().merge(passage);
        super.commit();
        errorMessage = "super.commit()";
    }catch(Exception e){
        super.rollback();
        throw new UserException(errorMessage, e.getCause());
    }

    return likeNum;
}

}

```

```

package com.me.dao;

```

```

import org.hibernate.Criteria;
import org.hibernate.HibernateException;
import org.hibernate.Query;
import org.hibernate.criterion.Restrictions;
import org.hibernate.mapping.List;

import com.me.exception.UserException;
import com.me.pojo.User;

```

```

public class UserDao extends DAO {

    public UserDao() {
    }

    //得到 UserID
    public User get(long userId) throws UserException {
        try {
            begin();
            Query q = getSession().createQuery("from User where userId = :userId");
            q.setLong("userId", userId);
            User user = (User) q.uniqueResult();
            commit();
            return user;
        } catch (HibernateException e) {
            rollback();
            throw new UserException("Could not get user " + userId, e);
        }
    }

    //注册 User
    public User register(User u) throws UserException {
        try {
            begin();
            getSession().save(u);
            commit();
            return u;
        } catch (HibernateException e) {
            rollback();
            throw new UserException("Exception while creating user: " + e.getMessage());
        }
    }

    //删除 User
    public void delete(User user) throws UserException {
        try {
            begin();
            getSession().delete(user);
            commit();
        } catch (HibernateException e) {
            rollback();
            throw new UserException("Could not delete user " + user.getName() , e);
        }
    }
}

```

```
}
```

```
//Login User
```

```
public User login(String name, String password) throws UserException{  
    try {  
        begin();  
        Criteria cri = getSession().createCriteria(User.class);  
        cri.add(Restrictions.eq("name", name));  
        if(cri.uniqueResult()==null) {  
            commit();  
            return null;  
        }  
        User user = (User)cri.uniqueResult();  
        if(user.getPassword().equals(password)) {  
            commit();  
            return user;  
        }else {  
            commit();  
            return null;  
        }  
    } catch (HibernateException e) {  
        rollback();  
        throw new UserException("Could not Search user " , e);  
    }  
}
```

```
//Get user Name
```

```
public User getUserName(String name) throws UserException {  
    try {  
        begin();  
        Query q = getSession().createQuery("from User where name = :name");  
        q.setString("name", name);  
        if(q.uniqueResult()==null) {  
            commit();  
            return null;  
        }  
        User user = (User) q.uniqueResult();  
        commit();  
        return user;  
    } catch (HibernateException e) {  
        rollback();  
        throw new UserException("Could not get user by name" + name, e);  
    }  
}
```

```

//Exist
public boolean isUserExist(String name) throws UserException {

    if(getUserName(name)!=null) {
        //存在返回 true
        return true;
    }
    return false;

}

}

```

## POJO

```

package com.me.pojo;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.PrimaryKeyJoinColumn;

@Entity
@PrimaryKeyJoinColumn(name="COMMENT_ID")
public class Comment {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long commentId;

    @Column(name = "content")
    private String content;

    @ManyToOne(targetEntity=com.me.pojo.User.class, cascade = CascadeType.MERGE)

```



```

        @JoinColumn(
            name="USER_ID",nullable=false)
        private User user;

        @ManyToOne(targetEntity=com.me.pojo.Passage.class,        cascade        =
CascadeType.MERGE)
        @JoinColumn(
            name="PASSAGE_ID",nullable=false)
        private Passage passage;

        public String getContent() {
            return content;
        }

        public void setContent(String content) {
            this.content = content;
        }

        public User getUser() {
            return user;
        }

        public void setUser(User user) {
            this.user = user;
        }

        public long getCommentId() {
            return commentId;
        }

        public void setCommentId(long commentId) {
            this.commentId = commentId;
        }

        public Passage getPassage() {
            return passage;
        }

        public void setPassage(Passage passage) {
            this.passage = passage;
        }

    }

```

```

package com.me.pojo;

import java.util.Set;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.JoinTable;
import javax.persistence.ManyToMany;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.OneToOne;
import javax.persistence.PrimaryKeyJoinColumn;

@Entity
@PrimaryKeyJoinColumn(name="PASSAGE_ID")
public class Passage {
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private long passageld;

    @Column(name = "title")
    private String title;

    @Column(name = "discription")
    private String discription;

    @Column(name = "content")
    private String content;

    @Column(name = "thumbNum")
    private int thumbNum;

    @ManyToOne(targetEntity=com.me.pojo.User.class, fetch = FetchType.LAZY, cascade =
CascadeType.ALL)
    @JoinColumn(name = "USER_ID",nullable=false)
    private User user;

```

```
    @OneToMany(targetEntity=com.me.pojo.Comment.class,fetch    =    FetchType.LAZY,  
cascade = CascadeType.ALL,mappedBy="passage")  
    private Set<Comment> commentCatalog;
```

```
    @ManyToMany  
    @JoinTable(name="Passage_User_Like")  
    public Set<User> likedByList;
```

```
    public Passage() {
```

```
}
```

```
    public Set<User> getLikedByList() {  
        return likedByList;  
    }
```

```
    public void setLikedByList(Set<User> likedByList) {  
        this.likedByList = likedByList;  
    }
```

```
    public String getContent() {  
        return content;  
    }
```

```
    public void setContent(String content) {  
        this.content = content;  
    }
```

```
    public long getPassageId() {  
        return passageId;  
    }  
    public void setPassageId(long id) {  
        this.passageId = id;  
    }  
    public String getTitle() {
```

```

        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getDiscription() {
        return discription;
    }
    public void setDiscription(String discription) {
        this.discription = discription;
    }
    public int getThumbNum() {
        return thumbNum;
    }
    public void setThumbNum(int thumbCount) {
        this.thumbNum = thumbCount;
    }
    public User getUser() {
        return user;
    }
    public void setUser(User user) {
        this.user = user;
    }
    public Set<Comment> getCommentCatelog() {
        return commentCatelog;
    }
    public void setCommentCatelog(Set<Comment> commentCatelog) {
        this.commentCatelog = commentCatelog;
    }
}

```

```
package com.me.pojo;
```

```
import java.util.HashSet;
import java.util.Set;
```

```
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
```

```
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.JoinTable;
import javax.persistence.ManyToMany;
import javax.persistence.OneToOne;
import javax.persistence.PrimaryKeyJoinColumn;
```

```
@Entity
```

```
@PrimaryKeyJoinColumn(name="USER_ID")
```

```
public class User {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private long userId;
```

```
    @Column(name = "name")
```

```
    private String name;
```

```
    @Column(name = "password")
```

```
    private String password;
```

```
    @Column(name = "email")
```

```
    private String email;
```

```
    @Column(name = "phoneNum")
```

```
    private int phoneNum;
```

```
    @OneToMany(targetEntity=com.me.pojo.Passage.class, mappedBy="user")
```

```
    //@JoinColumn(name = "USER_ID")
```

```
    private Set<Passage> passageCatalog;
```

```
    @OneToMany(targetEntity=com.me.pojo.Comment.class, cascade = CascadeType.ALL,
mappedBy="user")
```

```
    //@JoinColumn(name = "USER_ID")
```

```
    private Set<Comment> userComment;
```

```
    @ManyToMany(mappedBy="likedByList")
```

```
    //@JoinTable(name = "User_Passage", joinColumns = {@JoinColumn(name = "userId",
referencedColumnName = "userId")}, inverseJoinColumns = {@JoinColumn(name =
"passageId", referencedColumnName = "passageId")})
```

```
    public Set<Passage> userLikes;
```

```
public User() {
```

```
}
```

```
public Set<Comment> getUserComment() {
```

```
    return userComment;
```

```
}
```

```
public void setUserComment(Set<Comment> userComment) {
```

```
    this.userComment = userComment;
```

```
}
```

```
public Set<Passage> getUserLikes() {
```

```
    return userLikes;
```

```
}
```

```
public void setUserLikes(Set<Passage> userLikes) {
```

```
    this.userLikes = userLikes;
```

```
}
```

```
public void setPassageCatalog(Set<Passage> passageCatalog) {
```

```
    this.passageCatalog = passageCatalog;
```

```
}
```

```
public Set<Passage> getPassageCatalog() {
```

```
    return passageCatalog;
```

```
}
```

```
public long getUserId() {
```

```
    return userId;
```

```
}
```

```
public void setUserId(long userId) {
```

```

        this.userId = userId;
    }

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public int getPhoneNum() {
        return phoneNum;
    }
    public void setPhoneNum(int phoneNum) {
        this.phoneNum = phoneNum;
    }

    @Override
    public String toString() {
        return    name ;
    }

}

```