



院系 数据科学与计算机学院

学号 15352237 姓名 刘顺宇

学号 15352257 姓名 罗双

学号 15352263 姓名 米良

学号 15352278 姓名 齐新新

移动 Web 应用设计

大作业实验报告

【实验目的】

通过建立一个网站来综合应用本课程学习的知识。

【大作业分工】

组员分工可以按照模块（JSP 程序）分工，也可以按照软件开发过程分工，例如：系统设计、前台(html+CSS)、后台(JSP)、数据库管理、测试、安装、写文档。

【大作业评分标准】

- (1) 有无数据库增改删操作
- (2) 有无文件上传功能
- (3) 核心功能是否实现以及功能是否完整性
- (4) 项目的整体难度
- (5) 画面设计的美观
- (6) 实验报告编写
- (7) 提交时间

同一个组所有成员统一打分。所有 2~4 人的组，打分时不看人数，同等对待。对于单人 1 组的要有充足的理由（在实验体会中进行说明），否则要扣分。4 人以上一组的要求更多功能。

【实验环境】

tomcat, JDK 和 mysql。注意要允许外部访问应用程序需要[设置 Windows 防火墙](#)。

【实验要求】

- (1) 最好有数据库增改删操作，上传文件操作。例如：博客上传照片，增加修改删除博客。
- (2) 最好有管理员后台的操作功能，例如，浏览所有用户的博客内容，删除用户非法博客，某用户禁言。
- (3) 前后台都直接采用底层编程，**不要使用任何框架**。
- (4) 确定需求后，画出用例图（顶层）、类图、E-R 图，然后再编码。采用快速迭代的方法，先做一个小型完整的系统，有时间再不断增加功能或改进设计。
- (5) 整个项目完成后要在教学服务器上安装，安装说明由另一个文件给出。



[下面各项为要填写的内容]

【小组号】

15352237

【项目名称】

Higo 嗨狗旅游网站设计与实现

【项目描述】

1. 项目意义

网络技术日新月异的进步和互联网的日益普及，使人类的生活发生了巨大的变化。越来越多的企业单位和个人都在致力于发展新的宣传方式和媒介，同时为服务对象提供更加便捷和有效地服务。随着我国经济社会的迅速发展，人们的生活水平有了显著提高，假日经济和旅游经济已经成为人们消费的热点，各地也把旅游业作为本地经济发展的重要支柱之一，旅游网站也就成为了各地旅游胜地和旅行社对外宣传中不可缺少的工具，它的内容和设计对于旅游地的宣传和旅游项目的开发的决策起着非常显著的作用，也为消费者收集信息和旅游决策提供便利和至关重要的助力。在这种背景下，旅游和互联网的融合势必成为一种趋势，网络 IT 也将推动 IT 领域内电子商务的迅速发展，也为旅游业界注入新的活力。

结合本学期所学习的移动 web 应用设计课程内容，项目小组设计开发了 Higo 嗨狗旅游网站。通过项目的设计和实现，项目成员了解网站开发的流程及可能出现的问题，对于网站开发有了深入的了解与体会，也加强了课堂内容的理解与运用。

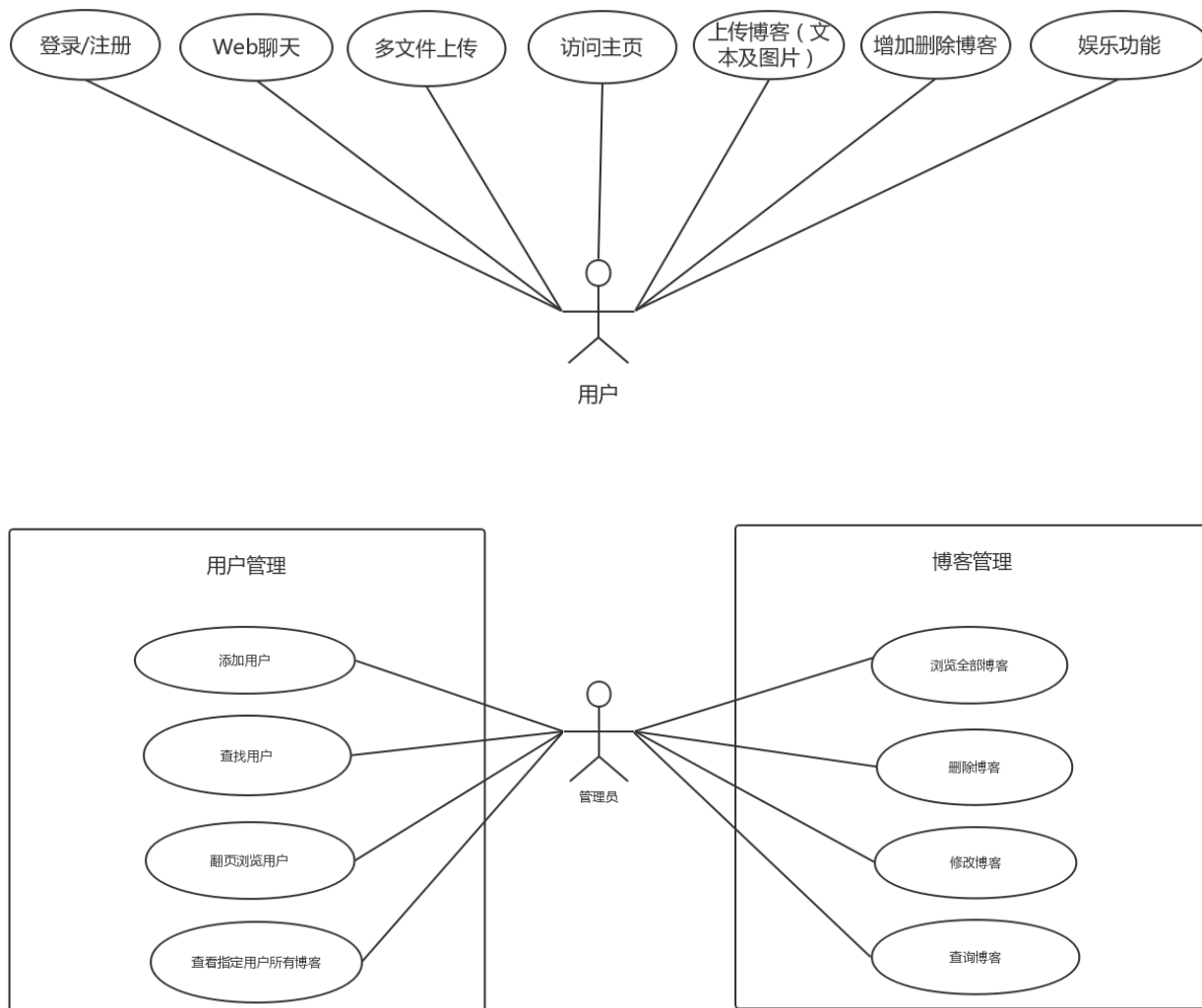
2. 项目功能

Higo 嗨狗旅游网站基于 JSP 动态网站开发的 web 应用程序，以互联网为传播媒介来实现旅游相关信息发布、共享的平台。本次实验报告主要论述了项目设计开发的基本过程、设计开发过程中主要运用的技术语言 and 平台及实现过程中遇到的问题和收获等。项目主要实现用户端注册和登录、博客平台文本和图片及多文件上传及删除、web 聊天功能，管理员管理包括查看新增用户、查找特定用户、禁言非法用户、浏览所有用户的博客内容和删除用户非法博客功能。



【系统设计】

1、顶层用例图

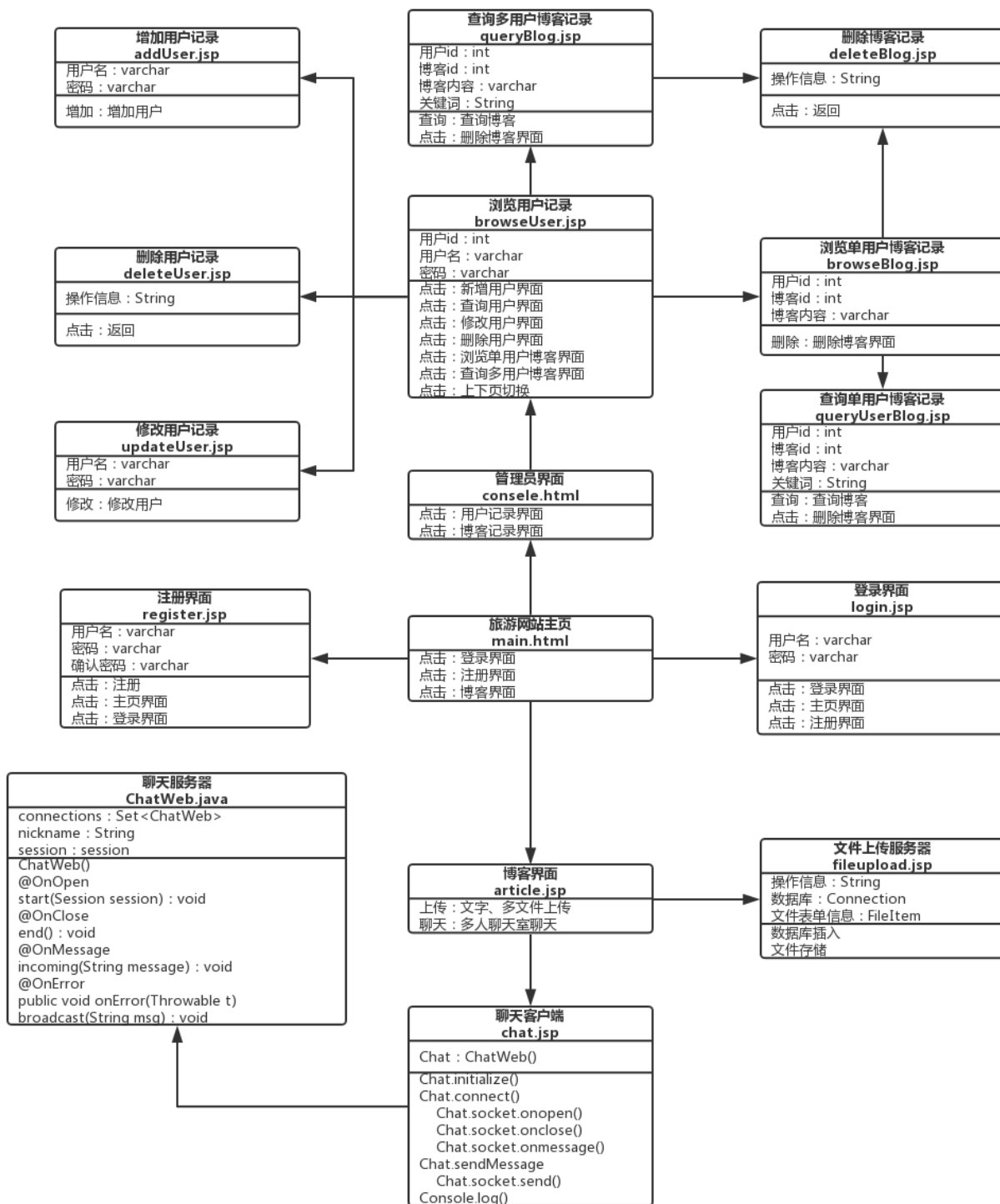




实验报告

2、类图

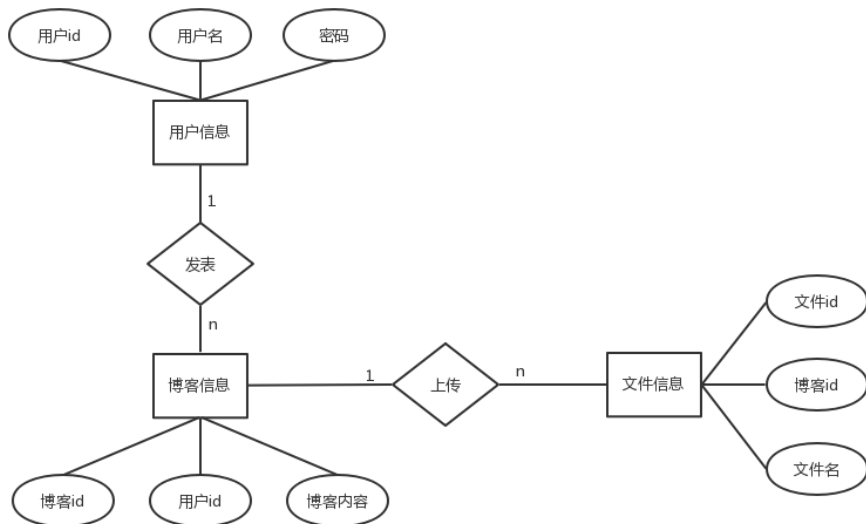
(可以把每个主要的 jsp 文件当成一个类)





3、E-R 图

(表之间的关系图)



4、数据库设计

表名: userinfo			
字段名	中文名	类型	说明
userid	用户 id	int	userid 为主键, 唯一标识每个用户, 自增长类型
username	用户名	varchar	具有唯一性约束, 即用户名不可重复
password	密码	varchar	每个用户名对应一个密码

表名: bloginfo			
字段名	中文名	类型	说明
blogid	博客 id	int	blogid 为主键, 唯一标识每条博客, 自增长类型
userid	用户 id	int	userid 为外键, 参考 userinfo 表中的 userid 列, 一个用户可以发多条博客, 删除用户时级联删除发表的博客
blogcontent	博客内容	varchar	用户发表的博客文字内容

表名: fileinfo			
字段名	中文名	类型	说明
fileid	文件 id	int	fileid 为主键, 唯一标识用户上传的每个文件, 自增长类型
blogid	博客 id	int	blogid 为外键, 参考 bloginfo 表中 blogid 列, 一条博客可以包含多个文件, 删除博客时级联删除发表的文件
filename	文件名	varchar	用户上传文件的名称, 如果是图像则会显示, 如果是其他文件则显示链接, 可下载

【运行截图】

(给出全部或主要操作画面的截图, 需要时可以在截图下面写一点简要说明)



主界面:



跳转博客页面

跳转登录界面

跳转注册页面

导航栏

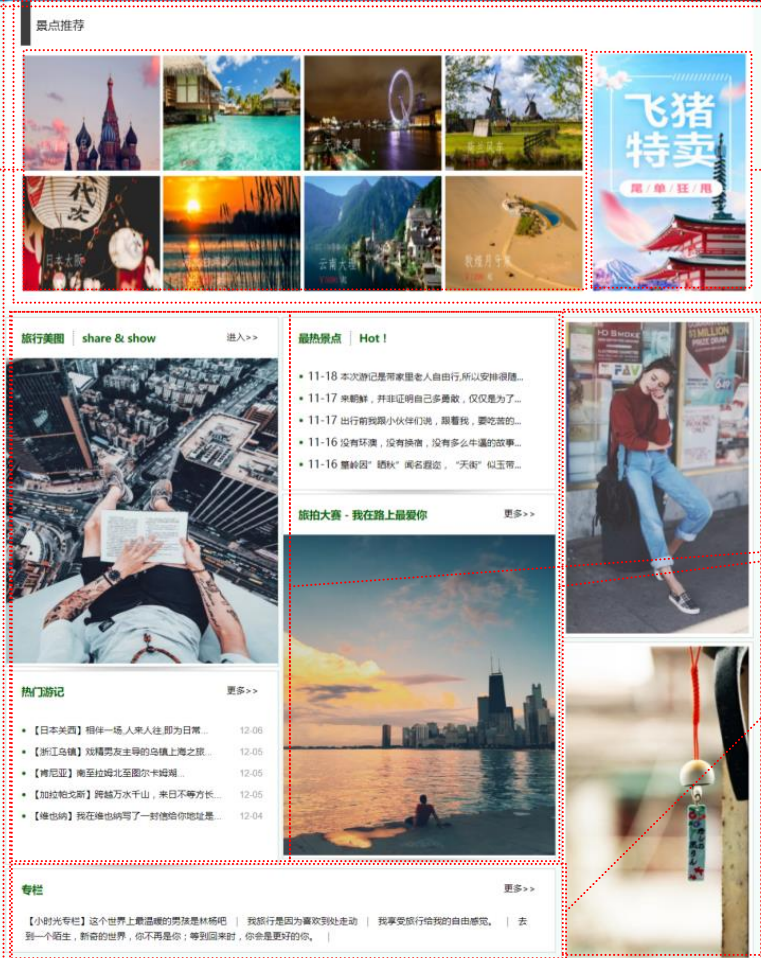
图片切换

点击图片
可实现 3D
切换

搜索框

景点推荐

广告栏



分栏

Higo! 带你去看全世界

页底

版权



中山大學
SUN YAT-SEN UNIVERSITY

实验报告

管理员界面：

Logo

跳转回主页

管理员账号

管理员主界面

跳转新建用户页面

跳转用户查询页面

跳转博客查询页面

跳转浏览用户界面

Home Page

首页

博客管理

查看消息

统计报表

访问总数
230,000,000

全站查询

查询

最新文章

Why Won't My Cat Eat?

Submitted by: Hiccup - June 10, 2012

Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non...

Approve

Decline

We Are About Color

Submitted by: Hiccup - June 10, 2012

Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non...

Approve

Decline

Ancient Technology

Submitted by: Hiccup - June 10, 2012

Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non...

Approve

Decline

Bird's Nest Soup

Submitted by: Hiccup - June 10, 2012

Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non...

Approve

Decline

最新注册用户

添加用户

查找用户

Squint
Front-End Engineer
18 minutes ago

Eunice
Architectural Designer
18 minutes ago

Captain Gutt
Software Engineer
18 minutes ago

Flynn
Accounting Manager
18 minutes ago

查看更多用户



注册界面:



h1 a: 点击可

连接到主页

Form 表单, 提交注

册信息到数据库,

并跳转到主页

链接: 可跳转到注

册界面或者主页

登录页面:



h1 a: 点击可

连接到主页

Form 表单, 提交登

录信息到数据库,

并跳转到主页

链接: 可跳转到注

册界面或者主页



博客页面：

header

m-arti form
发表文本提
交文件图片

main

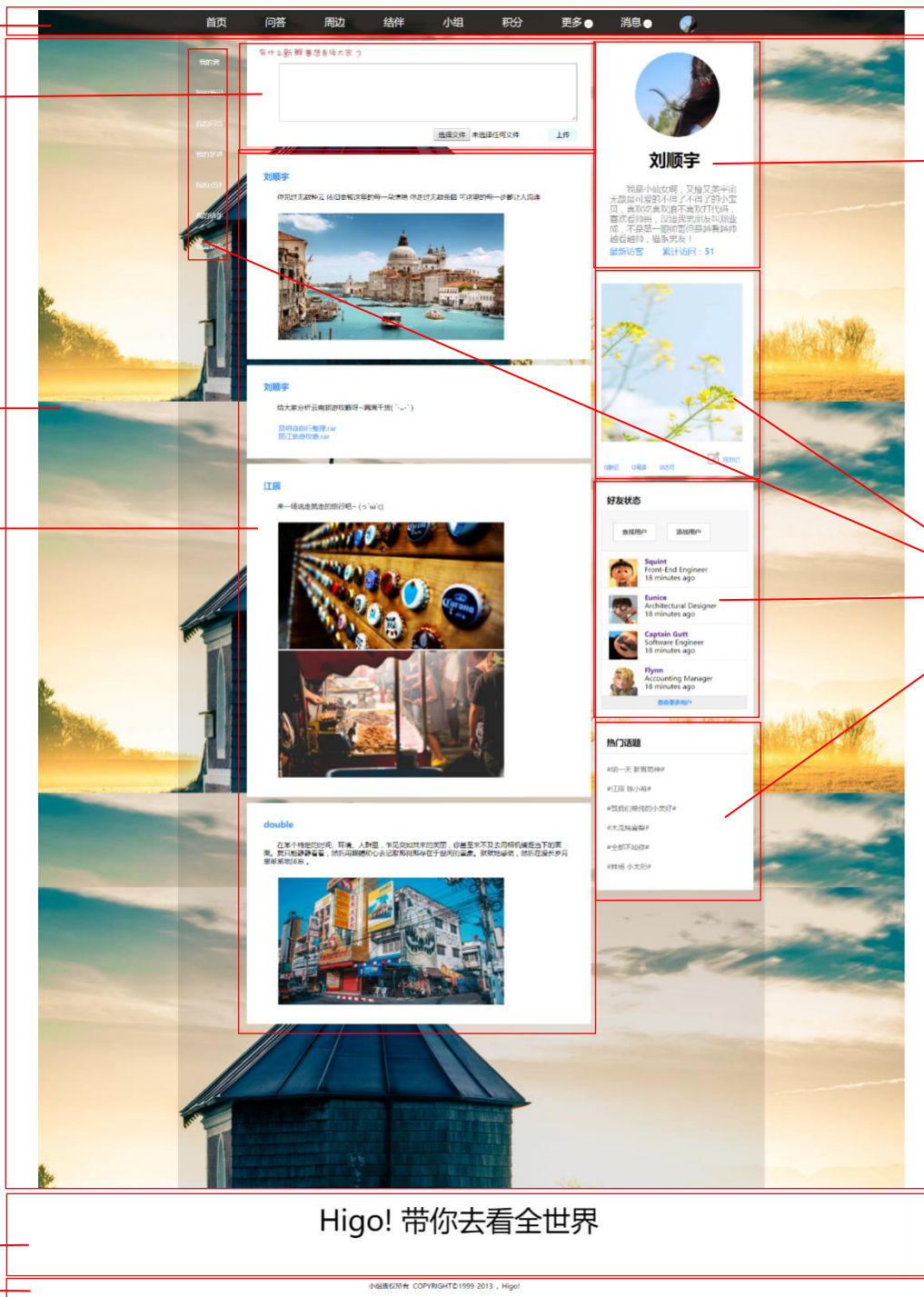
显示发表微
博，包括用
户名、文本、
图片或者上
传的文件链
接内容

用户信息：
用户名根据
登录的不同
用户而改变

各块级定义

footer

end





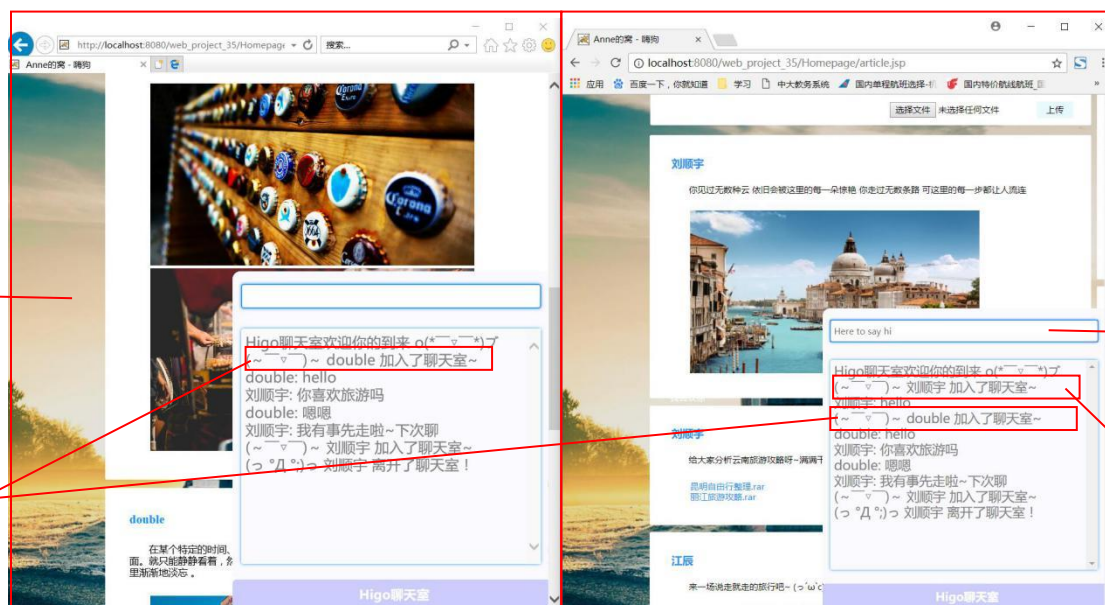
博客其他功能介绍：

web 网页版聊天室与嗨狗周边

初始界面：因为 chrome 浏览器截图会过滤掉聊天框，所以才用一般截图显示聊天室。



聊天界面：



嗨狗周边界面：网站 logo 的周边小作品，采用 PS 技术+css 动画效果展示效果，娱乐功能功能一部分，也是网站副产品的重要展示界面。





【实验体会】

写出实验过程中遇到的问题，解决方法和实验体会，每位同学单独写。

[刘顺宇 (15352237)]

在本次的项目过程中，我担任的工作是后端架构实现的工作，主要负责了 Web 聊天的实现、管理员数据库的搭建实现和一些其他的后台操作。

• 遇到问题及解决方法

1. WebSocket 通讯：在一开始尝试实现 web 聊天功能的时候查询资料试了多种办法，包括 ajax、servlet 等办法，最后终于用 HTML5 WebSocket 实现了多人聊天室的功能。接下来我们从 WebSocket 通讯的三个阶段（打开握手、数据传递、关闭握手）以及 WebSocket 客户端与服务器两个角度对 WebSocket 聊天实现的代码进行解析

1.1. WebSocket 客户端：websocket 允许通过 JavaScript 建立与远程服务器的连接，从而实现客户端与服务器间双向的通信，而要实现连接我们需要按照以下的步骤。

- 申请一个 WebSocket 对象：参数是需要连接的服务器端的地址，同 http 协议使用 http://开头一样，WebSocket 协议的 URL 使用 ws://开头，另外安全的 WebSocket 协议使用 wss://开头。websocket 的 url 开头是 ws，如果需要 ssl 加密可以使用 wss，当我们调用 websocket 的构造方法构建一个 websocket 对象（new WebSocket(url)）的之后，就可以进行即时通信了。

```
var url = 'ws://' + window.location.host + '/web_project_35/chat';  
websocket = new WebSocket(url);
```

- WebSocket 对象定义了几个监听函数：一共支持四个消息 onopen, onmessage, onclose 和 onerror，我们可以看出所有的操作都是采用消息的方式触发的，这样就不会阻塞 UI，使得 UI 有更快的响应时间，得到更好的用户体验。

当 Browser 和 WebSocketServer 连接成功后，会触发 onopen 消息；

```
websocket.onopen = function(evt) {};
```

如果连接失败，发送、接收数据失败或者处理数据出现错误，browser 会触发 onerror 消息；

```
websocket.onerror = function(evt) {};
```

当 Browser 接收到 WebSocketServer 发送过来的数据时，就会触发 onmessage 消息，参数 evt 中包含 server 传输过来的数据；

```
websocket.onmessage = function(evt) {};
```

当 Browser 接收到 WebSocketServer 端发送的关闭连接请求时，就会触发 onclose 消息。

```
websocket.onclose = function(evt) {};
```

1.2. WebSocket 服务器端：将一个普通 Java 对象（POJO）使用 @ServerEndpoint 注释作为 WebSocket 服务器的端点。，@ServerEndpoint("/chat")表示将 WebSocket 服务端运行在 ws://[Server 端 IP 或域名]:[Server 端口]/项目/chat 的访问端点，客户端浏览器已经可以对 WebSocket 客户端 API 发起 HTTP 长连接了。

在建立连接之后我们通过四种注解函数来进行服务器操作：

- @onMessage 注解的 Java 方法用于接收传入的 WebSocket 信息。
- @OnOpen 在这个端点一个新的连接建立时被调用。
- @OnClose 在连接被终止时调用。
- @OnError 在发生错误时被调用



2. WebSocket 在建立连接时实现传参功能：在实现 web 聊天时，希望想实现在建立连接时客户端能传一个参数即自己的用户名给服务器，而服务器得到用户名之后能一直保留并用以聊天。一开始尝试使用通过 @PathParam 获取页面传值但是失败了，后来直接使用 send() 传递参数并标注便实现了这个功能，下面简单介绍一下两种办法。

1.1. 通过 @PathParam 获取页面传值：在页面与 Java 后台建立连接的时候获取页面上提供的参数，也就是在 @OnOpen 注解的方法里面获取一次页面的参数，可以使用 @PathParam 注解实现。

- 在被 websocket 映射的 Java 类中的注解如下：

```
@ServerEndpoint("/chat/{userId}/{userName}")
```

其中 chat 是映射地址，/{userId}/{userName} 分别为页面可以传入的两个参数

- 在该 Java 类中的方法中：

```
@OnOpen
public void onOpen(@PathParam("userId") String userId,
                   @PathParam("userName") int userName, Session session)
```

这样就可以获取第一步中的两个参数的值(userId 和 userName)

- 在 javascript 脚本中

```
Chat.initialize= function() {
    if(window.location.protocol == 'http:') {
        Chat.connect('ws://' + window.location.host + '/web_project_35/chat');
    } else {
        Chat.connect('wss://' + window.location.host + '/web_project_35/chat');
    }
};
```

1.2. 直接在 onOpen() 时使用 send() 函数来传递参数，在 java 类中使用一个标记符来记录此时是不是第一次接收信息，是的话进行相应处理。

```
Chat.socket.onOpen= function() {
    var username = "<%=current_user_name%>";
    Chat.socket.send(username);
};
```

• 实验感想

这次项目我主要负责后台的代码架构搭建与实现，从一开始实现 web 聊天，到后面的数据库设计，以及各种小细节的设计，虽然一开始构思比较简单，但是在具体实现起来的时候也遇到了重重困难，上网查询了很多资料，咨询了同学、TA 和老师，最后才解决了问题。花费最多时间的应该就是 web 聊天方面，具体的问题以及实现思路也在上面提及了，在数据库方面的设计我主要负责协助工作，协助另一个队友一起完成了一些数据库操作方面的代码。

总的来说，这次实验从一开始有一个雏形，到后来的实现，都离不开队友们的辛勤付出，这次的团队真心太给力了，所有的需求只要定好了基本都马上实现，执行力特别强，特别感谢这次队友们的给力输出，最后的实现效果真的是特别的棒。希望自己以后也能再接再厉，继续加油！



[罗双 (15352257)]

• 遇到的问题及解决方法

在本次实验中，我负责的主要任务是整体项目数据库的设计和登录和注册页面连接数据库以及博客界面的发表博客并可以同时上传多个文件。登录和注册页面和数据库的连接比较容易，因为在之前的实验中也做过相关的数据库操作，只是这一次的设计需要考虑更多的情况，比如输入用户名或为空，密码输入错误，用户名不存在以及注册时密码和确认密码不一致等细节问题需要考虑清楚，才能使得和用户的交互比较友好。

已经做完之后才发现原来上述很多功能都可以通过 js 实现，相比于 jsp 会更加简便，看来还需要继续学习。比较难的是多文件上传，因为上网查的很多关于 ajax 实现多文件上传的教程都使用了框架，比较少底层实现，因此做起来比较难，而且不太熟悉 js 的语法，比如 click 事件触发和监听器等操作，学了比较久，所幸最后成功实现了多文件上传，还是很开心的。实验过程注意遇到了如下问题：

1. 因为在用户发表博客后，需要在下面显示用户名和发表的博客内容以及上传的文件等，可是其中关联了多个表，因此需要执行多次 `ResultSet rs=stmt.executeQuery()` 查询，一开始做的时候，是一个 stmt 来进行多个 rs 操作，但是显示一直失败，浏览器提示 `ResultSet 已经关闭的错误`，后来经过查询资料得知执行完一条 rs 操作后，rs 就会被关闭掉，因此无法执行下一条查询。一个 stmt 最好对应一个 rs，如果用一个时间内用一个 stmt 打开两个 rs 同时操作，就会出现上述情况。后来便通过多创建几个 stmt，一个 stmt 对应一个 rs 解决了。

2. 关于数据库的设计，一共有三个表，用户信息、博客信息和文件信息，因为一个用户可以发表多条博客，一条博客可以上传多个文件，所以在博客信息中定义了用户 id 作为外键，参考用户信息中的用户 id，然后文件信息中定义了博客 id 作为外键，参考博客信息中的博客 id。一开始设计的时候，没有考虑周全，导致后来在做后台管理员界面的时候，想删除用户，但是因为其发表了博客，导致无法删除，后来重建了新的数据库，设置了外键绑定关系中的级联更新和级联删除，这样在删除用户的时候，可以将其关联的博客和文件全都删除。可见在最初设计数据库的时候需要将整体考虑的比较完整，不然重建还是比较麻烦的。

3. 在用户发表微博的时候，一开始设计是用户先上传文件，文件上传成功后再点击发布后将文件和博客内容发布出来，上传文件是链接到 `fileupload.jsp` 中进行上传，点击发布时是自己提交给自己。可是在文件信息中，需要博客 id 作为外键，标识每个文件属于哪个博客，但是上传文件后还没有 blogid，只有在发布后才会生成自增长的博客 id，导致上传文件失败。后来经过思考，觉得可以文字和文件一同提交给 `fileupload.jsp`，然后在其中加一个判断语句，如果是表单字段，则通过 session 将用户 id 取出，再将输入的内容 `blogcontent` 取出来后插入语句，插入到博客信息表中，这样就生成了一个博客 id 了，然后再判断如果是文件的话就将其放到指定路径后执行对文件信息的插入操作，将博客 id 和文件名记录进数据库。

4. 多文件上传使用的是 XMLHttpRequest 上传文件，大致的流程是在 `fileSelected()` 中监听选择文件信息，然后通过 file API 上传文件。File 对象可以获取上传文件的信息，还可以读取这个文件的内容。其中，File 对象是来自 body 中在一个 `<input>` 元素上选择文件后返回的 id 为 filename 对象。然后在 `uploadFile()` 函数中关联表单数据，进行事情监听，如果没发生异常的话则跳转到 jsp 文件进行上传，在 jsp 中的内容和之前学过的文件上传类似，通过 while 循环将每个文件取出然后放入指定路径。

• 实验感想

这次大作业确实花了比较多时间完成，因为大家都希望实现较多的功能，中途也遇到了大大小小的各种问题，后来和队友一起慢慢解决了。对于后台应用的实现，一开始就应该有一个比较清晰的架构，还有对 jsp 的执行顺序以及参数传递都需要比较了解才能做得比较流畅。感觉这次关于 js 的部分应用还是比较少，看了下现在后台大部分都是 js 实现的，相比于 jsp 会更加便捷，因此还是需要不断学习。

做后台的工作其实挺有趣的，每次实现新功能时成就感满满，但确实需要比较细心，然后对语法啥的比较好理解才行。总而言之，队友都完成的非常出色，界面设计的很好看，今后会继续努力！



[米良 (15352263)]

• 遇到问题及解决方法

在本次项目过程中，我担任的工作是登录、注册、微博及嗨狗周界面设计。

因为任务的特殊性，界面涉及到后台相关设计，所以我是小组最先开始做的人，所以，也面临着和其他成员不一样的问题——底层界面的架构，为整个网页代码分块，定义不同的类名和 ID 名。开始时很混乱，后来参考了以前做过的练习实验和一些网站源码，如蜂窝窝、腾讯视频等等，在不断修改中也顺利渡过了第一阶段。

登录和注册界面很相似，所以只要注意设计出一个另一个就很好的解决了。在这两个界面中遇到的最大的问题大概就是 button 的样式设计，默认的实在是真的是太丑了灰灰的，设计过程中一直都有注册按钮的边框，后来查了以前 PPT 发现了 button 这个元素，加上 `background-color:rgba(146,250,212,.8); border: none;` 这两条样式就成功去掉了边框修改了背景颜色。

微博界面一开始参考了蜂窝窝的个人界面，后来发现功能太少二栏式界面就显得很单薄，就采用了三栏式几乎重新写了整个界面，好在最终的结果是可以的。实验过程中主要遇到了如下问题：

1. 表单提交的模块无法点击聚焦。通过 chrome 的 debug 功能查看网页和代码模块定义的关系，发现微博发布的表模块覆盖了表单提交模块，最开始通过设置表单提交模块优先级大于微博发布模块的方法，这个时候发现虽然解决了表单部分被覆盖的问题，但是出现了 header 的下拉菜单被遮挡的情况，并且这个时候设置 header 菜单的优先级也并没有起作用。最后通过调整整个界面模块的布局顺序（先编写向右浮动的模块，再编写中间的微博发布和表单两个模块），将表单模块放在了微博内容发布模块的后面，就成功解决了这个问题，同时也方便了后期三列式的布局定位。

2. 后期加入后台程序的工作时，发现在不同电脑上界面的分布会出现错位。经过多次尝试，修改 relative 定位的 left 和 top 值用百分比表示，就顺利解决了不同浏览器的兼容问题。

3. 后期合并过程中发现在缩小浏览器窗口的情况下，背景左侧会出现空白。尽管采用了填充整个 container 的方法，但是还是不行。后来修改代码放在 body 里，用图片填充背景，设置溢出隐藏，不随鼠标滑动移动，问题解决。

嗨狗周界面设计一开始并不在我们的计划中，后来阴差阳错就做了一个简陋的界面。吸取微博展示界面 px 单位的教训，图片大小特意用百分比限制，但是也出现了点小问题：

1. 在给图片采用 relative 定位时，`float:left;` 会使图片大小发生改变。采用 absolute 定位时，会使图片的 top 位置发生改变（这个肯定的），所以最终还是采取 relative 定位方式，用 `left: 100%;` 定位部分图片到界面右侧，但同时又出现了下面的问题。
2. 背景采用颜色渐变效果，在设置图片位置为 position 定位时，发现会随着图片 `left: 100%;` 的设置出现背景横向拓展。调整参数发现会根据 left 的设置大小产生不同的拓展。通过增加样式 `body {overflow: hidden;}` 很好的解决了问题（前提：`width: 100%`）。没有查到原因，猜测是块级元素定位导致背景宽度溢出显示界面，所以宽度被拉长，设置溢出部分被隐藏就可以了。

• 实验感想

对于底层界面的架构，这要求代码的结构要很清晰，注意不同版块之间的区别与联系，据此编写不同的类名 ID 名，同时定义的名字要清楚易于理解。发现虽然难度并不高，但是真的是一个需要清洗的逻辑和耐心的工作。后来和一些师兄师姐聊天得知，一些大的网站都是采用框架或者是套用模板和代码生成器一键生成的，总算是对前端工作拾起了一点信心。

实验过程中遇到了很多问题，大大小小很多，因为时间跨度有点长数量太多已经记不太清。嗨狗周界面的设计是因为小组同学在后期美化界面时修改主界面菜单栏背景为黑色，需要祛除 logo 图的背景，又担心 logo 大面积黑色会影响 logo 效果，所以扣了图之后又把 logo 的狗的衣服颜色改成了红色，想想又同时换了彩虹色，又再一顺手给它 P 了一个灯笼一个元宝上去，想想图不能浪费啊，一个小组成员正遗憾我们页面设计没有用到动画，我就又顺手写了个简陋的周界面展示产品，也为了纯属为看到我们界面的人一



个消遣吧。

因为这半年看了挺多 UI 方面的书籍和设计，也做了一些作业，所以这次对界面设计要求稍稍有一点点高，尽量系统的去美化整套界面。再加上队友的高要求和自己的强迫症以及高清的超级漂亮的图的加持，最后的界面效果还是很满意的。各种大神队友实现了挺多功能，作为旅游网站的一些基本功能都已经实现。在整个实验的过程中也充分利用了老师教授的知识，除此之外也补充了其他的干货和技巧，体验了 web 开发的流程，对于整个网站有了很深刻的认识，可以说收获很多很多。

[齐新新 (15352278)]

• 遇到问题及解决方法

在本次项目过程中，我担任的工作是主页页面设计及后台管理员主页以及子页面设计。

在主页页面设计中，主页头部包括导航栏、图片切换以及搜索框；内容页面分上下两栏，上栏分左右两栏，左栏为景点推荐，右边为广告推送；下栏分左右两栏，分栏效果如截图所示。在主页页面设计中，遇到了如下问题：

1. 设置导航栏处于浏览器界面最上面的部分时，发现无论怎么设置，导航栏总是下移 20px：

在发现了这个问题之后，检查了很多遍 header 的 css 语句，并没有发现问题（也有可能是有语句没有填写），就暂时搁置了这个问题，后来在测试不同部分的范围的时候显示每一个 div 的 border，导航栏却自动移到浏览器界面最上面，虽然还不是很清楚原理是什么，但是这个问题在我后来修改了 header 结构的时候仍然可以用显示 border 边界的方法解决，足以证明这是一个好办法；

2. 图片切换没有使用之前上课做的 html 的方法，因为我觉得过于繁琐，所以选用了 js，但开始学 js 并没有那么轻松，设置 addeventlistener 的时候因为不懂原理所以卡了很长时间，导致在图片切换总是定时切换，当鼠标悬停到图片上的时候图片切换不能停止。另外，又因为图片切换是 3D 切换，所以在旋转的时候会超出 header 部分，侵入到下面 mainbody 中，因此此处需要设置 overflow:hidden，就可以在不改变 3D 切换的效果的情况下隐藏图片超出部分。

3. 设计好主页页面后，我把控制不同部分的 css 语句分到不同的 css 文件中，但是由于在设计 id 或者 class 的时候没有规范命名，有一些命名是重复的，所以就会出现命名一切都是正常的，但是在 link 一个 css 文件之后，整个页面就会自动左移，原来的结构全部乱掉，后来发现是在 footer 里面有重名的 id，在修改的时候不小心对整个页面做了设置。

在后台管理员页面设计中，管理员头部包括网站 Logo、返回主页按钮、管理员账户信息；下一栏包括首页、博客管理、查看消息以及统计数据，但是为了查询方便，只能在首页界面点击，无法访问其他功能（这里主要是因为会用到一下 ajax 和 jquery 的响应方面的语句，所以没有添加）。后天管理员界面遇到的问题比较少，主要包括以下两个：

1. 在管理员查看数据库的界面，因为使用的是 form 提交到 jsp 文件，但是通常需要设置两个按钮，一个是查询按钮，另一个是返回按钮，当查询按钮按下，表单就会自动提交到对应的 jsp，但是如果此时将返回按钮的连接

2. 在查询页面和删除博客或者用户页面，都应该输出<msg>信息，但是输出的文字太丑了所以开始我删掉了，但是如果不输出的话，不能得知是不是查询成功或者删除成功，所以在做的时候数据库连接失败了也不知道，这其实是一个非常不用户友好的管理员界面，所以添加了回来，但是如果想做的好看，可以使用 js 写一个 alert 语句，不用直接显示在浏览器页面上会好很多。

• 实验感想

这次项目选择了前台的界面设计，但是界面设计并没有我想象的那么简单，一个简单的 3D 图片切换因为之前没有学过 js 花了很长时间，所幸最后实现的效果不错也没有白费，主页其他部分的内瓢，是借鉴之前中大网页，结合我们旅游网站的主题而设计的，现在看来美感和技术点兼具还是不错的，不过单凭我



自己是找不到这么好看图片，还是多靠队友的审美。至于后台管理员界面设计，其实应该还是比较复杂的，单一个后台管理员系统都可以独立成为一个大项目，所以我并没有做的很具体，只是简单的有一个比较好看的界面并且能实现管理员对后台的数据库操作而已，想要更进一步，还需要 ajax 和 jquery 的学习，希望会在之后的专业课学习中涉及到这些知识点。

总结一下这次 project，整个 project 完成下来，界面设计确实用心了，其次，队友真的很给力，没有一点拖沓或者不认真，而且最后呈现的效果其实远在我意料之外，同时这次真的学到了课堂之外的一些知识，对前端开发的理解又深了一点，总而言之，非常开心！

【提交报告】

每个小组交一份实验报告。

截止时间：2017 年 12 月 16 日 23:00 (周六)

上传网站：<http://172.18.187.11/netdisk/default.aspx?vm=15web>

实验上交/大作业/实验报告

上传文件：

- (1) 小组号_大作业_实验报告.doc
- (2) 小组号_大作业_组员分工.xlsx
- (3) 小组号_大作业.rar (所有源程序文件)