JSP综合大作业项目报告

1. 开发背景：

现在的社会，已然是网络科技高速发展的时代，每一个行业乃至一个小小的个体都在无时无刻的跟网络接触，在信息技术高速发展的当今，人们逐渐开启互联网自我营销的时代。随着学习的不断进阶，在老师的一步一步带领下，逐渐的开始步入实战项目的大厅，历时一个学期的学习，终于开始了自己独立的项目开发实战部署，随着老师一声令下，各个小组团队的纷纷开始 策划自己小组的项目实施方案，终于，我选择了 个人博客后台内容管理系统（CMS），在此之前，从来没有过独立开发一个完整的网站，有的只是 静态网站的搭建，现在，热血沸腾的时候到来了，我要开始实施我的 项目了。

1. 项目分析：
   1. 项目目标：

（个人博客）内管理系统（CMS）的搭建是信息时代自媒体的载体。通过（个人博客）内容管理系统，一个小小的个体也能够轻轻松松的开启自我营销的大门，通过使用（个人博客）内容管理系统，个人可以轻松的 发表文章、发布相册、管理网站的留言和评论、并且后期可以轻松的实现额外功能的增添。通过该系统，每一个人都可很快的在网络上自我营销.在网站的建设过程中，将注重应用体系的健壮。

应用体系：主要体现在信息的发布上使信息能够方便、快捷、

可靠、及时、准确的发布；以及网站的信息的高效管理

* 1. 用户定义

开发的内容管理系统，主要针对的用户群就是个人，每个人只要有

意愿在网络大流中对自身进行自我营销，那么这个系统都将使用与这些用户。

* 1. 交付平台

本地window系列机子测试（若上线，则将放于Linux服务器环境）

* 1. 项目的开发环境：

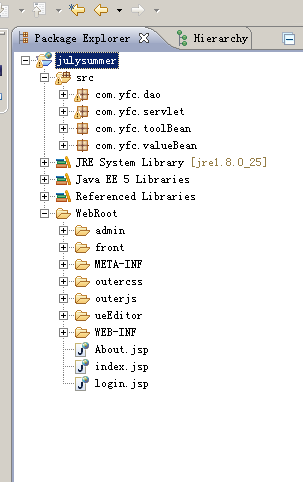
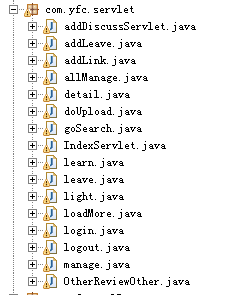
这次环境使用的学校的电脑进行开发

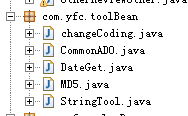
开发工具为：myeclipse8.5/myeclipse2014 tomcat6.x jdk 8/jdk 6

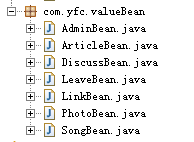
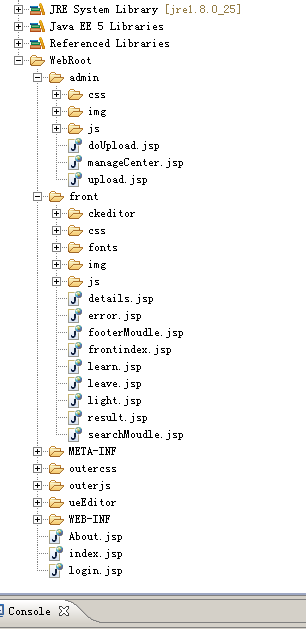
此外我发现，不能老是切换环境开发，因为我当初-频繁的切换开发环境，导致项目的日志里产生了很多对项目有影响的东西，从而影响了我的项目的开发。

* 1. 内容目录分工说明

下图中是我的项目的代码目录截图：

在上面的内容目录中，其中的分工为:

包 com.yfc.servlet :负责接受并处理前台发送的请求，负责页面间的

跳转，调用其他bean来完成请求的处理

包 com.yfc.dao : 专门与数据库打交道，负责各个功能模块的数据

的增、删、改等的功能。

包 com.yfc.valueBean: 每一个类都对应了数据库中的每一个表，主要

负责，对数据进行封装，方便数据的传递.

包 com.yfc.toolBean: 该包内的类主要起辅助性的作用，如日期格式

的转换，编码格式的转换等。

WebRoot下的文件：

文件夹 admin：对应的前台管理员的界面的jsp 文件和各种js

文件和css 文件等

文件夹 front： 对应的是前台的访客界面的jsp文件和各种js文件

和css文件等。

文件outercss、outerjs文件：对应的是某个页面独立的css和js 文件

文件 ueEditor文件：是后台管理员文章编辑插件

* 1. 信息框架与流程图：

管理员

网站后台

友情链接管理

博主管理

站内搜索

文章评论管理

留言管理

相册管理

文章管理

查看

查看

删除

删除

更新

添加

删除

删除

更新，删除，添加

根 据 标 题 模 糊 匹 配

添加

删除

查看

增加

网站访客

网站后台

查看博主信息

站内搜索

查看相册

发表留言

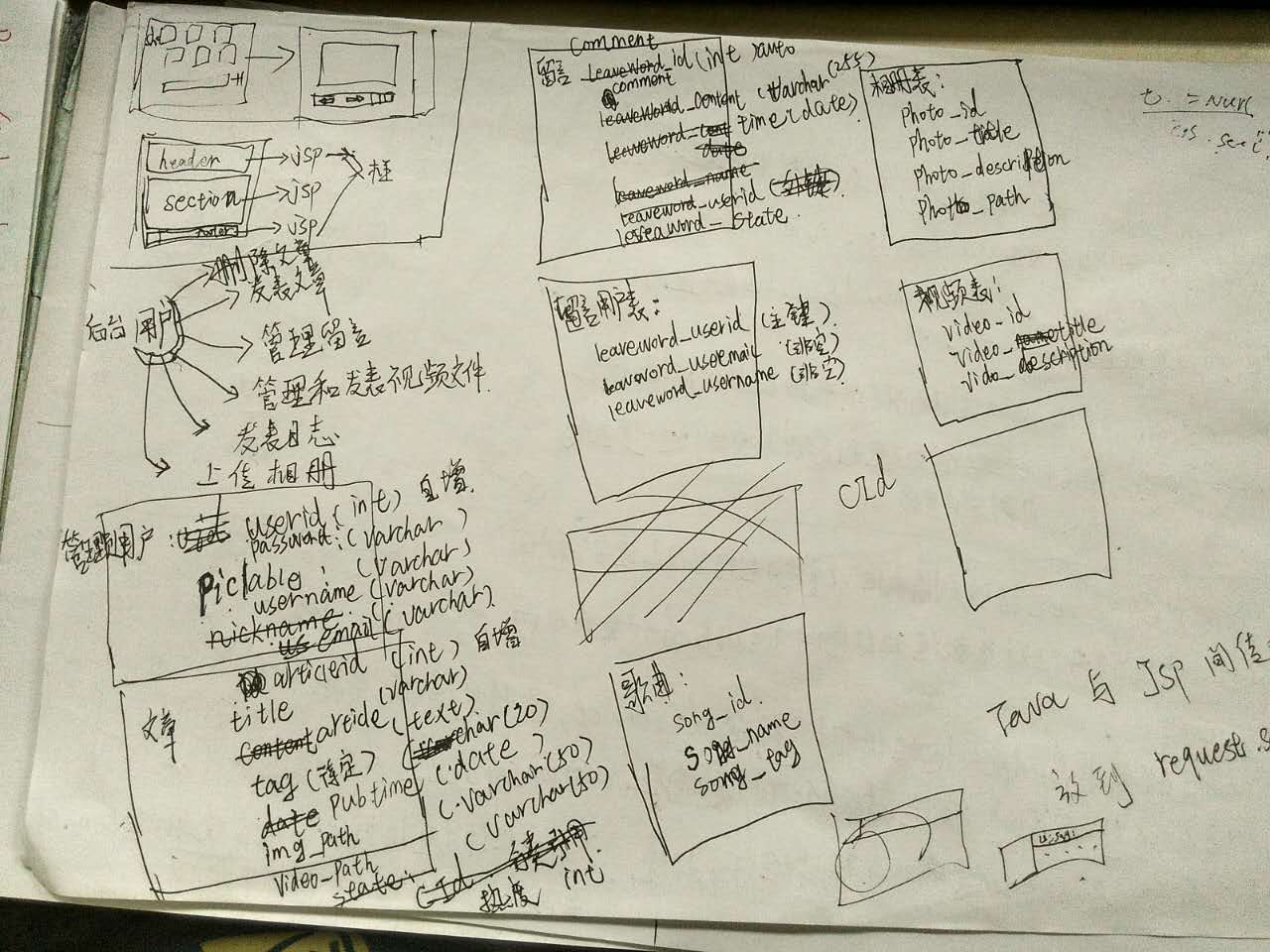
评论文章

查看文章

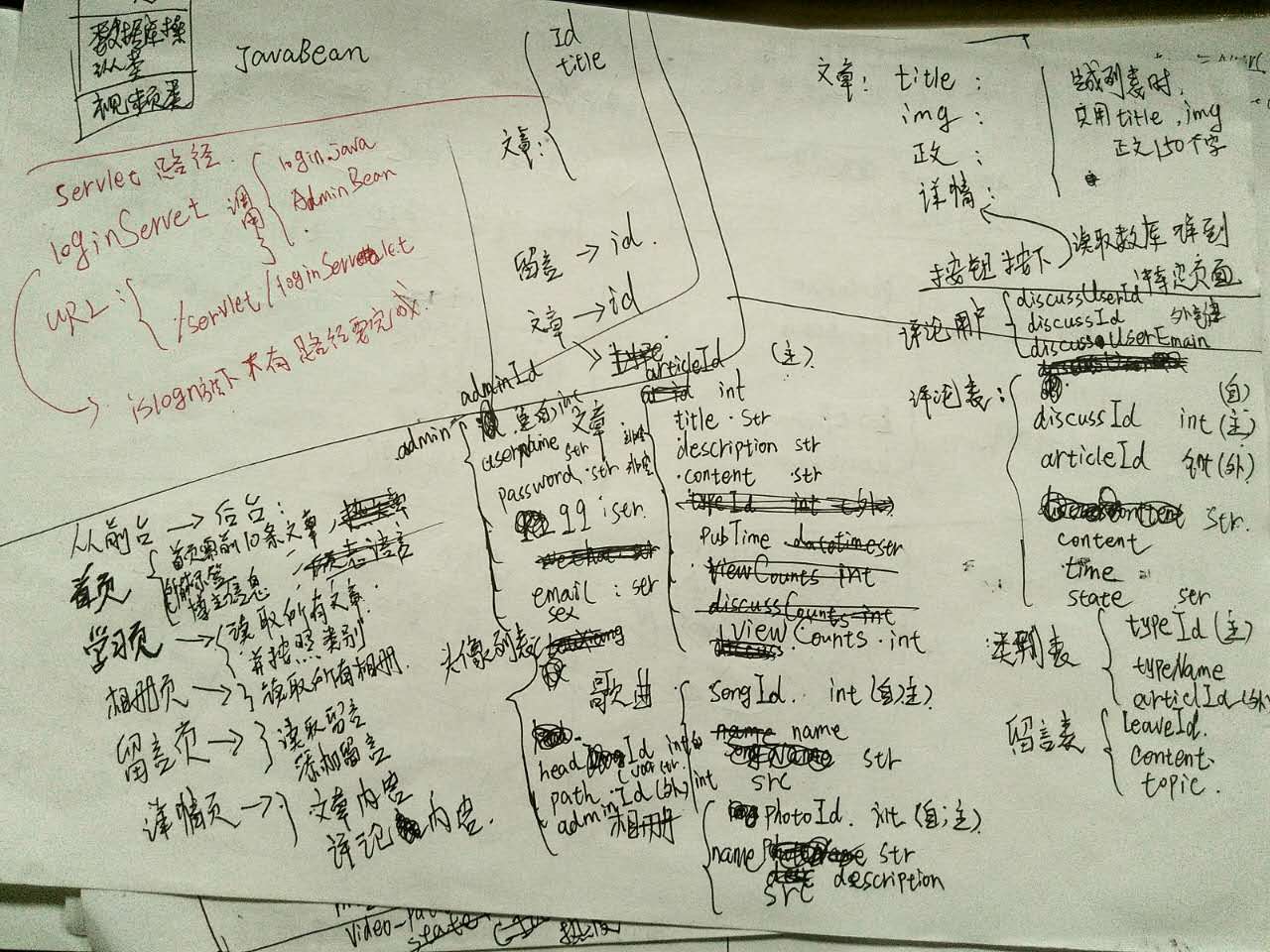
1. 课程开发记录：
   1. 课程设计构思过程：

“（个人博客）内容管理系统” 从设计到完成、从无到有都是自己一点一滴构思而来。为什么要选（个人博客）呢容管理系统 这个课题呢？我记得一句话“简单的也可以做的不简单。”，而且在之前我就很想实现一个完整的个人博客网站，但由于一直没有入手后台的实现，总没有成功，所以，在这次上了JSP的课程后，我便毅然决然的选择了这个课题。因为是一个人做这个项目，所以在整个项目的前后台都得自己设计并实现，在构思的过程中，对于 数据的设计 和 前端页面 的设计可谓耗费时间占整个项目开发时长的近半，在设计好数据库和前端的布局之后，我首先是根据计划好的的功能进行了后台代码的编写，在后台代码编写好后，便开始了前台代码的编写和优化。

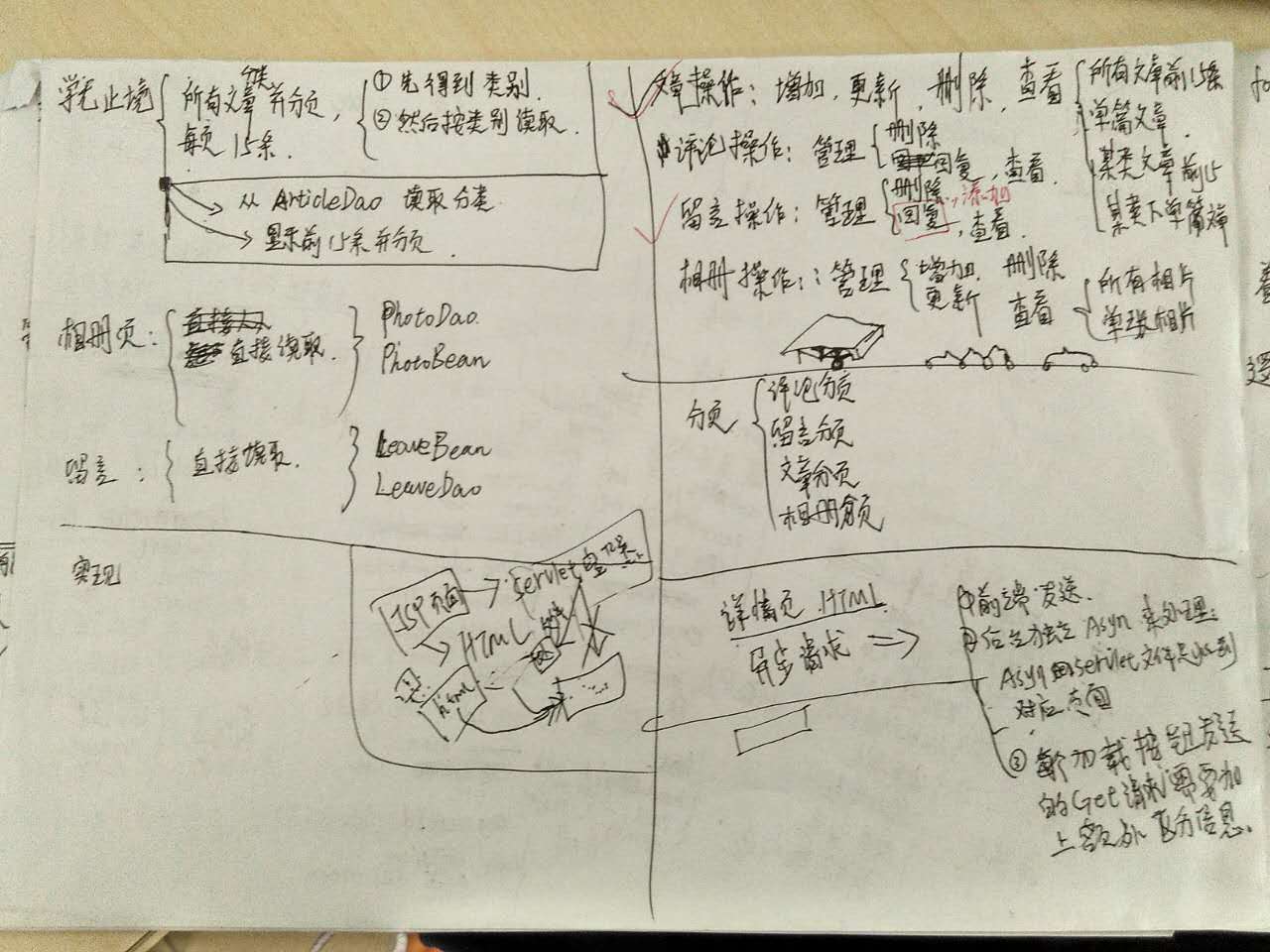
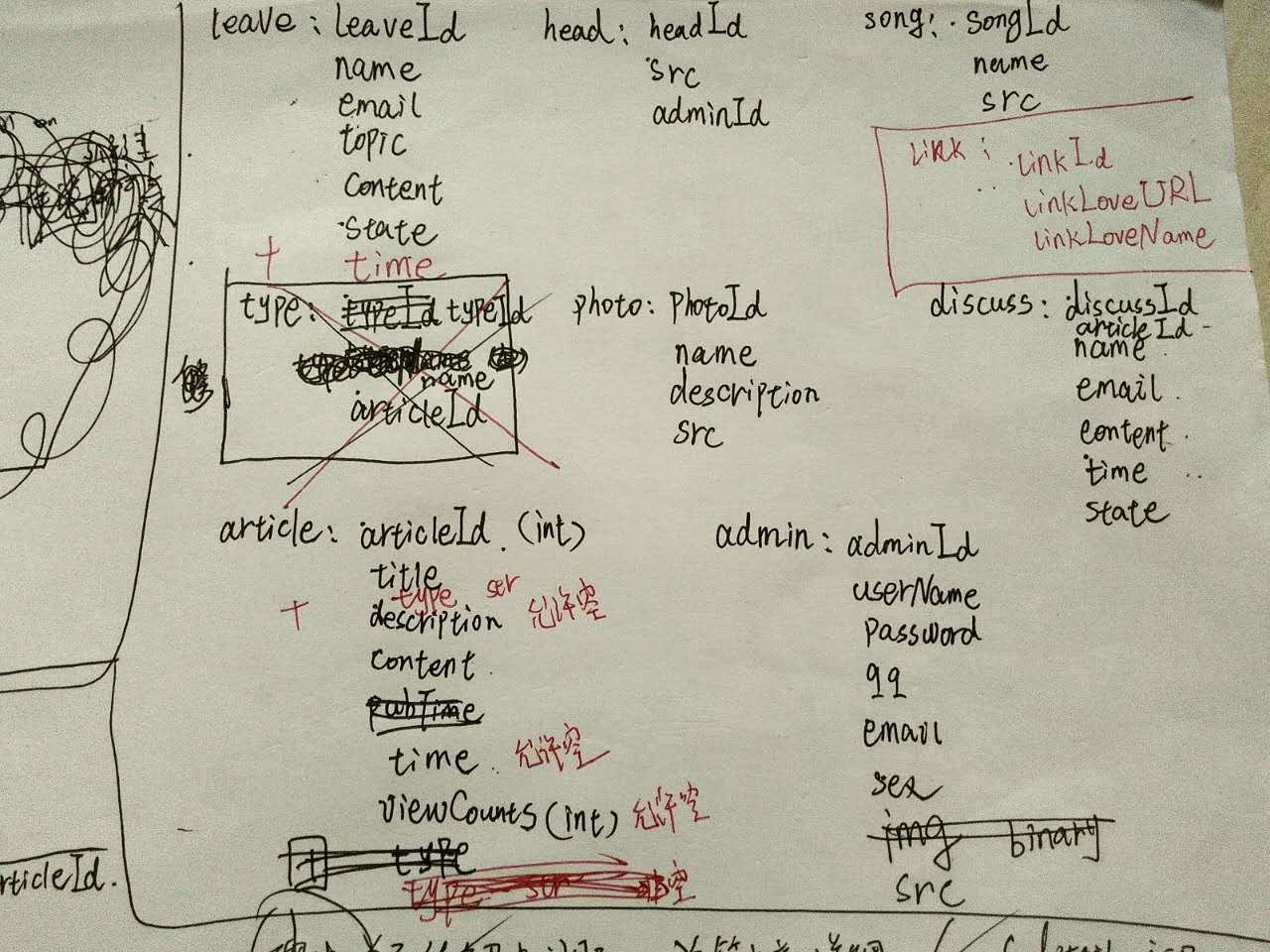
下图为我设计之初的草稿图：



早期设计



中期设计



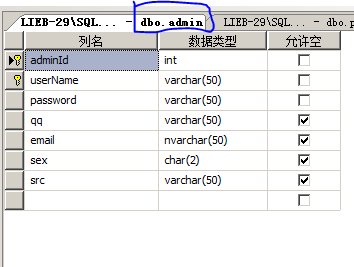
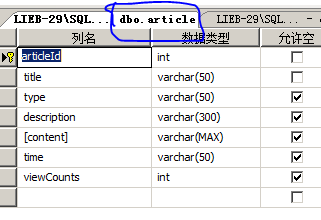
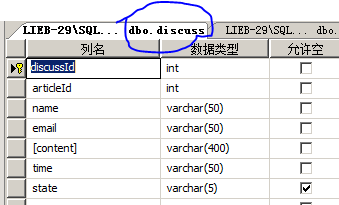
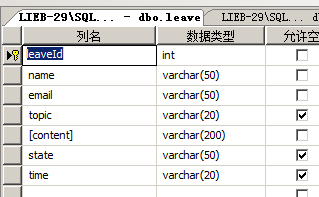
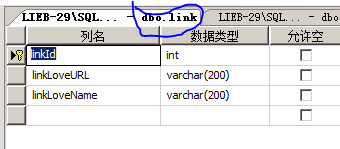
最终设计

最终设计

* 1. 制作过程简介：

在整个系统的设计过程中，最重要的就是数据库的设计，一旦数据库设计不好，那么接下来的编码，便会很痛苦，一系列的问题将会导致代码大的更改。下面是我经过几次更改后最终的数据库设计结果。

数据库设计：（如下图：）



数据库设计完了，接着便是该如何去架构我们的代码，是的各个代码之间的耦合性不至于太强，于是，我采用了老师推荐的MVC模式，也是当下较流行的设计模式，

系统设计模式：

采用了MVC 模式设计

JavaBean (M)

数据库

JSP页面（V）

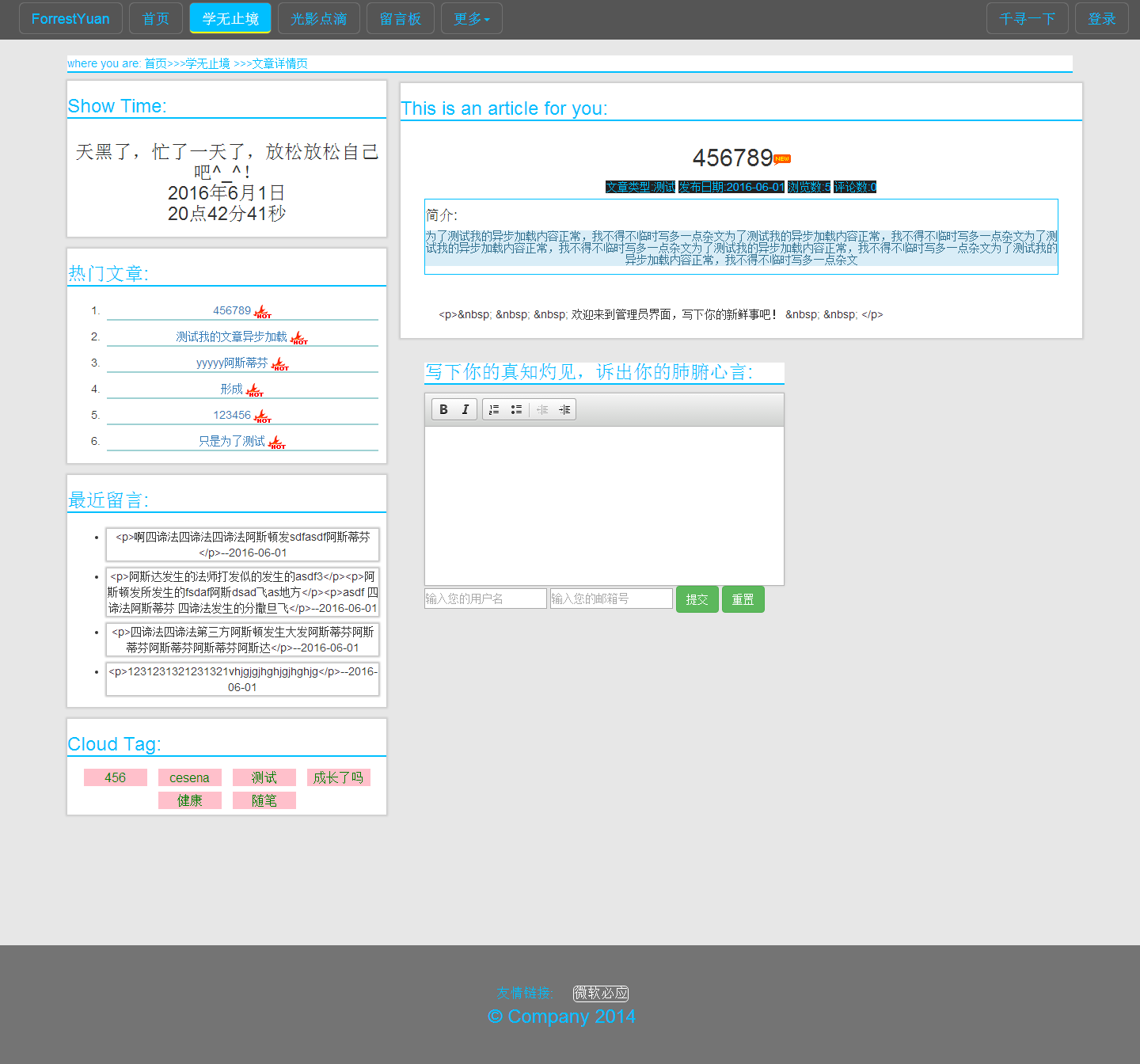
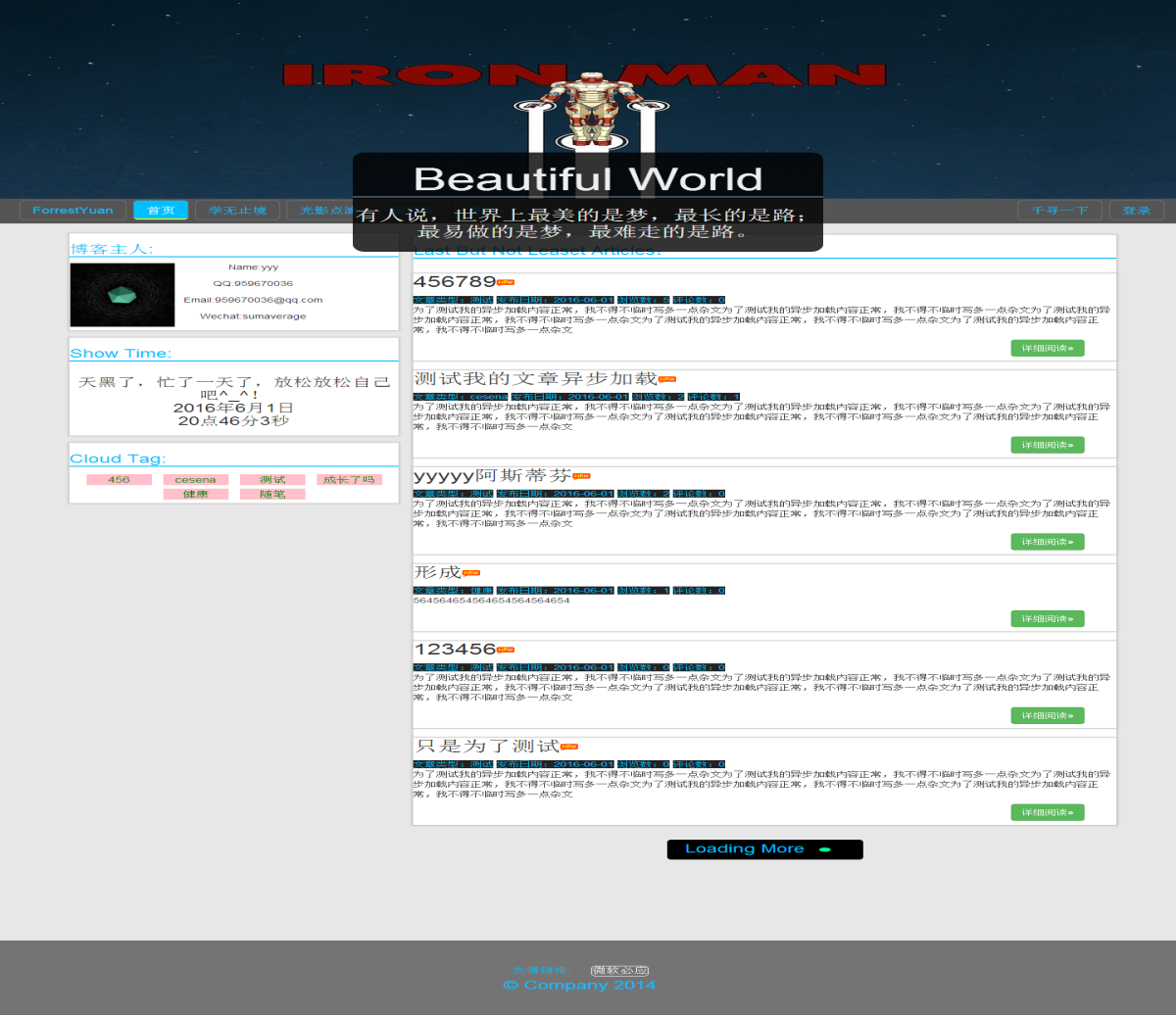
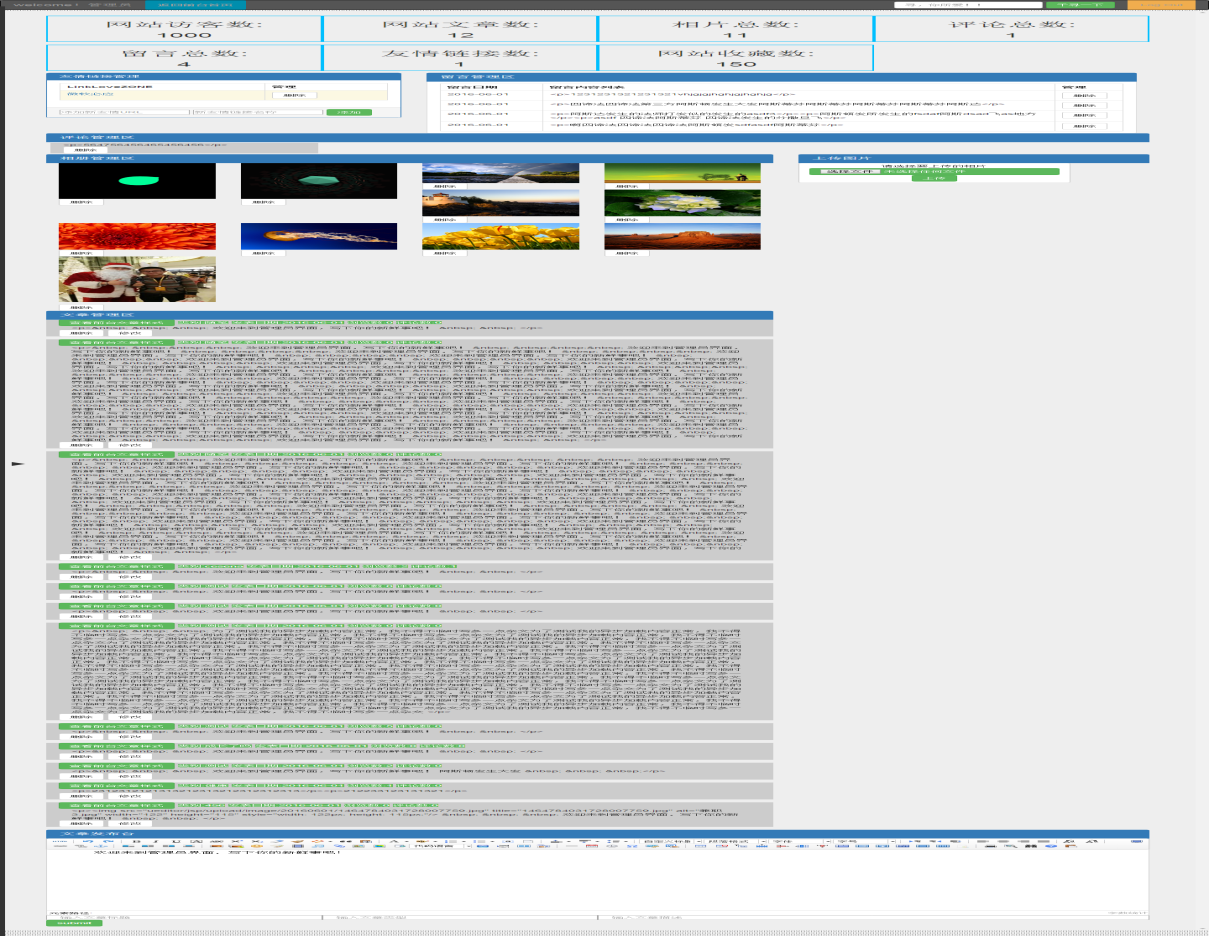
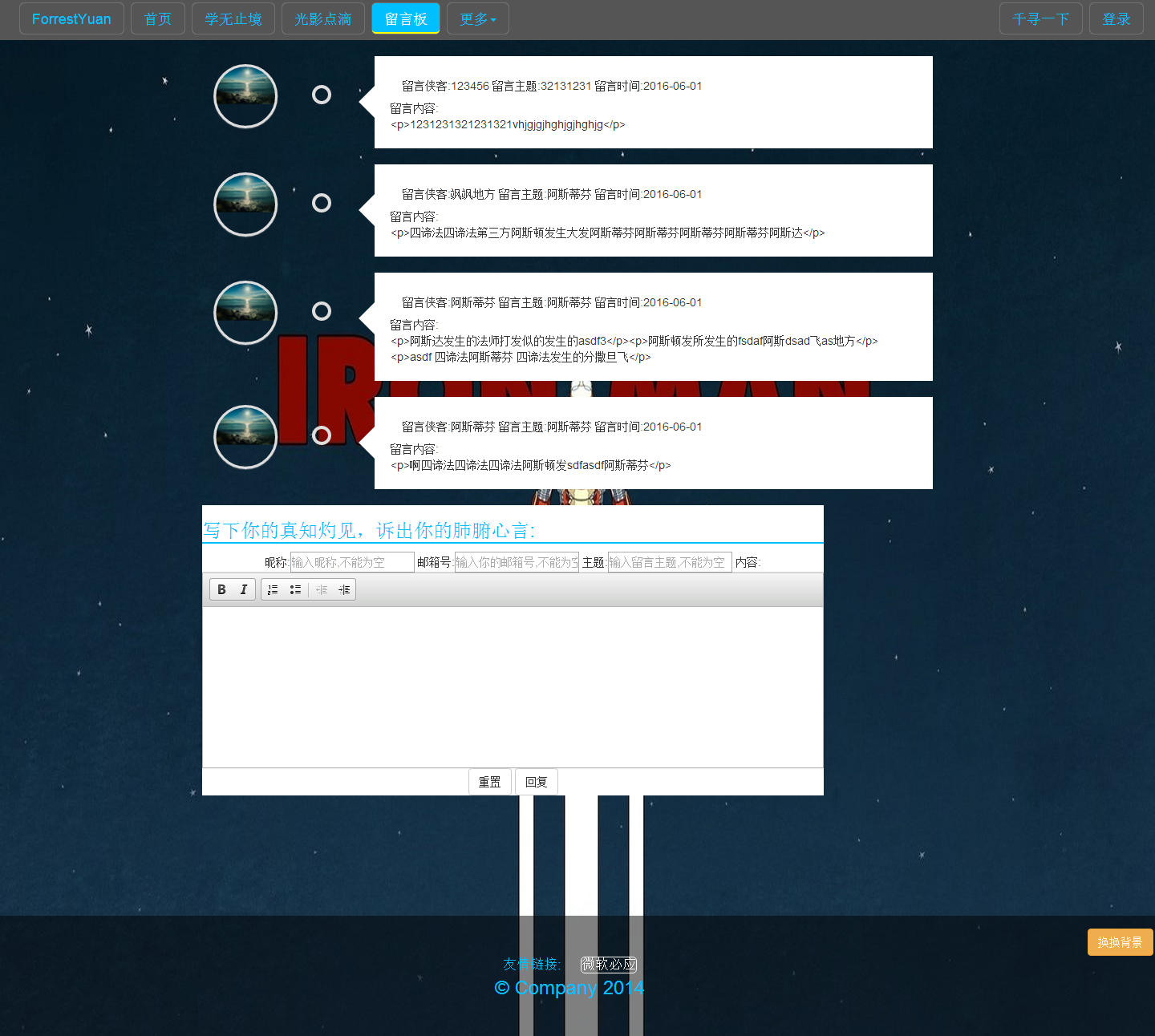
Servlet (C)

界面设计：

对于前端的设计，主要采用现在最流行的BootStrap框架，实现了界面的个浏览器的较强的兼容性，以及移动设备优先的特点，在使用bootstrap的框架下，再自己进行相应的布局设计，且利用css3和 jquery的特性给网站增加一些动态交互效果，从而实现界面友好的前台。

界面如下图所示（有些动态效果需要在浏览器里才能体验到）：

图片是直接截下来的，所以图片看起来的效果并没有直接在浏览器看好。



某些代码展示：

代码的编写严格按照相应的编码规范进行编写。例如对于类的成员函数都采用了文章注释，方便在日后调用该成员函数时明白该函数的功能是什么。

下面给出的是代码的片段，仅供参考：

package com.yfc.dao;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import com.yfc.toolBean.CommonADO;

import com.yfc.valueBean.ArticleBean;

public class ArticleDao {

private CommonADO commonADO=null;

public ArticleDao(){

commonADO=CommonADO.getCommonADO();

}

/\*\*

\* @功能 实现对文章进行增、删、改的操作

\* @参数 oper为一个String类型变量，用来表示要进行的操作；single为ArticleBean类对象，用来存储某个文章的信息

\* @返回值 boolean型值

\*/

public boolean operationArticle(String oper, ArticleBean single) {

/\* 生成SQL语句 \*/

String sql = null;

oper=oper.toLowerCase().trim();

if(oper.equals("add")){//发表新文章

sql = "insert into article (title,type,description,content,time,viewCounts) values('"+single.getTitle

()+"','"+single.getType()+"','"+single.getDescription()+"','"+single.getContent()+"','"+single.getTime()+"',"+single.getViewCounts()+")";

}

if(oper.equals("modify")){//修改文章

sql = "update article set title='"+ single.getTitle()+"',type='"+single.getType

()+"',description='"+single.getDescription()+"',content='"+single.getContent()+"',time='"+single.getTime()+"',viewCounts="+single.getViewCounts

()+"where articleId=" + single.getArticleId();

}

if(oper.equals("delete")){//删除文章

sql = "delete from article where articleId="+single.getArticleId();

}

if(oper.equals("readtimes")){//累加阅读次数

sql = "update article set viewCounts=viewCounts+1 where articleId="+single.getArticleId();

}

/\* 执行SQL语句 \*/

return commonADO.executeUpdate(sql)>=1;

}

/\*\*

\* @功能 实现查询：文章下的阅读量最多的前指定条文章

\* @返回值 文章列表 list

\*/

public List<ArticleBean> queryHotArticle(int num){

List<ArticleBean> HotArticleList=new ArrayList<ArticleBean>();

String sql="select top "+num+" \* from article order by viewCounts desc";

ResultSet rs=commonADO.executeSelect(sql);

try{

while(rs.next()){

ArticleBean atb=new ArticleBean();

atb.setArticleId(rs.getInt("articleId"));

atb.setTitle(rs.getString("title"));

atb.setType(rs.getString("type"));

atb.setDescription(rs.getString("description"));

atb.setContent(rs.getString("content"));

atb.setTime(rs.getString("time"));

atb.setViewCounts(rs.getInt("viewCounts"));

HotArticleList.add(atb);

//System.out.println("我在articleADO 下，检查是否有数据："+atb.getDescription());

}

}catch(Exception e){

e.printStackTrace();

}

return HotArticleList;

}

/\*\*

\* @功能 实现查询：所有文章、所有文章下的单篇文章、某类别的前15条文章 、某类别下的单篇文章文章

\* @参数 id 为一个String类型变量,type为String类型变量

\* @返回值 文章列表 list

\*/

public List<ArticleBean> queryArticle(String id,String type){

List<ArticleBean> articleList=new ArrayList<ArticleBean>();

String sql=null;

if(id==null&&type==null){

//System.out.println("当前测试的是所有文章");

sql="select \* from article";

}else if(id!=null&&type==null){

//System.out.println("当前测试的是、所有文章下的单篇文章");

sql="select \* from article where articleId="+Integer.parseInt(id);

}else if(id==null&&type!=null){

//System.out.println("当前测试的是、某类别的前15条文章 ");

sql="select top 15 \* from article where type='"+type+"'";

}else{

//System.out.println("当前测试的是、某类别下的单篇文章文章");

sql="select \* from article where articleId="+Integer.parseInt(id)+"and type='"+type+"'";

}

try{

ResultSet rs=commonADO.executeSelect(sql);

while(rs.next()){

ArticleBean atb=new ArticleBean();

atb.setArticleId(rs.getInt("articleId"));

atb.setTitle(rs.getString("title"));

atb.setType(rs.getString("type"));

atb.setDescription(rs.getString("description"));

atb.setContent(rs.getString("content"));

atb.setTime(rs.getString("time"));

atb.setViewCounts(rs.getInt("viewCounts"));

//添加到articlelist中

articleList.add(atb);

}

}catch(Exception e){

e.printStackTrace();

}

return articleList;

}

/\*\*

\* @功能 搜索文章

\* @参数 str为String类型变量

\* @返回值 文章列表 list

\*/

public List<ArticleBean> getTosearch(String str){

List<ArticleBean> articleList=new ArrayList<ArticleBean>();

String sql="select \* from article where title like '%"+str+"%'";

ResultSet rs=commonADO.executeSelect(sql);

try{

while(rs.next()){

System.out.println("asdfasdf");

ArticleBean atb=new ArticleBean();

atb.setArticleId(rs.getInt("articleId"));

atb.setTitle(rs.getString("title"));

atb.setType(rs.getString("type"));

atb.setDescription(rs.getString("description"));

atb.setContent(rs.getString("content"));

atb.setTime(rs.getString("time"));

atb.setViewCounts(rs.getInt("viewCounts"));

articleList.add(atb);

}

}catch(Exception e){

e.printStackTrace();

}

return articleList;

}

/\*\*

\* @功能 查询文章总数

\* @返回值 int型变量

\*/

public int articleAll(){

String sql="select count(articleId) as count from article";

int counts=0;

try{

ResultSet rs=commonADO.executeSelect(sql);

if(rs.next()){

counts=rs.getInt("count");

}

}catch(Exception e){

e.printStackTrace();

}

return counts;

}

/\*\*

\* @功能 查询所有标签

\* @返回值 标签List

\*/

public List<ArticleBean> queryArticleType(){

List<ArticleBean> typeList=new ArrayList<ArticleBean>();

String sql="select count(articleId),type from article group by type";

try{

ResultSet rs=commonADO.executeSelect(sql);

while(rs.next()){

ArticleBean atb=new ArticleBean();

atb.setType(rs.getString("type"));

//添加到articlelist中

typeList.add(atb);

}

}catch(Exception e){

e.printStackTrace();

}

return typeList;

}

}

1. 个人课程设计总结

在开发这个系统之前，刚开始构思这个项目的时候，我对系统该如何去架构真的是一点思路都没有，我也都有去网上了解代码的架构，但我都是无法真的明白其意，不过还好老师最后讲解了一些实战项目的代码，让我毛塞顿开，明白了该如何去架构我的代码，而且很幸运，经过两个星期多的开发，终于完成了这个项目，虽然还是存在很多的不足点，但我起码从中学到了、收获了很多。还有，我其实感觉自己就是老师说的那个很傻的人，因为起初，老师建议我们尽量使用现有已有代码，在开发代码之前，我不以为意，我一直坚持认为，既然要写自己的一个系统，那么所有的代码都都应该自己写，比如对数据库操纵的代码，老师都有提供给我们ＤＢｕｔｉｌ这个类，可我在进行项目开发时都没有用上去，因为我觉得自己在学习阶段，尽量自己编写，更能锻炼自己的代码编写能力，所以我也就这么做了，可是，很快缺点就暴露了，我写的代码并不那么如意，而且代码的缜密性没那么强，所以我也意识到了，学会去使用他人的代码也是一个很重要的能力之一。

在开发系统过程城中，因为前台的发表评论和发表留言，以及后台管理员的发表文章用的编辑器，因为不想用html的textarea控件，所以就用了ckeditor（前台）和uEditor（后台）可是，可能没有去了解这些编辑插件的特性，所以导致了输出的结果没达到预期的效果，还有另外一个遇到的就是 编码格式问题，应为网络传输协议规定的传输格式与我的开发环境编码格式不统一，所以，也会经常遇到乱码的问题，但还好老师都有在课堂上给我们讲解过这个问题的解决办法，所以总的来说整个系统的开发过程还算是流畅的，唯一的不足之处就是，因为超过90%的代码都是我自己一码一码敲出来的，而我只有一个人来编写代码，累得很阿！另外， 我对我自己的项目不满意的地方就是，安全性和没有跟多复杂的功能，（能力不足的问题吧！），实用起来还是很有困难的。

不过还好，我真的从这次项目开发中学到了很多，想MVC的真正实用起来、dao层的应用、MD5的应用等等。

开发最后，进行项目演示，我当时就出现了搜索功能失败的原因，课上一直没解决，后来，我在排错的过程中发现，原来是由于我过于频繁的切换我的开发环境，从而导致了项目中产生了很多不同环境下的开发日志，导致我的代码都无法被调用。（真是悲催呀）关键时刻掉了链子。不过通过展示我们开发的项目，也是对我们学生的成果的很好的肯定（虽然或许做的并不好。），讲解自己的项目也是一个能力的体现，我还是很喜欢这种模式，真正的锻炼到了我的能力.

1. 参考资料

在开发这个系统过程中，100%自己开发一个完整的项目是很难的。因为很多功能的实现仅仅靠自己在这有限的开发周期内完成几乎是很难很难的，因为可能前人完成这个功能而花费的时间是好今年甚至更久，所以，想要在有限的时间内开发一个完整的项目，就像我们老师说的那样，得学会用别人的代码，转化成自己的代码，在应用到自己的项目中去。因为我开发的项目较为简单，所以引用的代码也就较少，但还是有的.所以在我的项目中参考如下资料：

1、前端的实现便是用了 bootstrap 的框架。

2、密码的加密 采用了MD5 的加密

3、前端和后台的编辑功能分别采用了 ckeditor和ueditor

4、文件上传功能参考了（实现的比我的要好）参考了图书馆某本jsp的书籍

1. 成绩（自我评分）

从我的整个项目来看，我基本上满意（毕竟是我的第一个自编自码的项目）。所以我对自己的评分: 86-96之间吧

1. 评语（自我评价）

通过这次项目的开发，我对自己的评价是：有一句话说的很好：“生如逆旅，一苇以航，凡心所向，素履所往。”对于这次的项目的开发，我还是很认真的对待的，因为，我真的很想尝试自己独立去开发一个项目。所以，无论我的项目最终是否真的很OK，我都很满意自己的项目，因为它能够正常运行。最后，我对自己评价是：我其实真的很不错的^\_^。