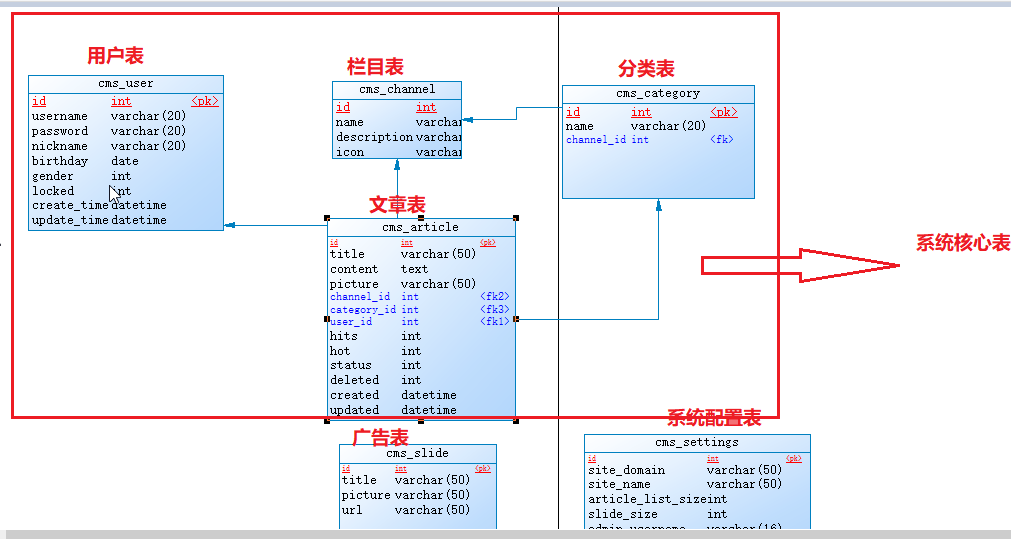
CMS系统开发过程

# 表结果关系图



## 用户表 cms\_user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段说明 | 字段类型 | 长度 | 为空 | 备注 |
| id | 主键id | Integer |  | 否 | 唯一主键，自增长 |
| username | 用户名 | String | 50 | 否 | 校验唯一性 |
| password | 密码 | String | 32 | 否 | **MD5加密** |
| nickname | 昵称 | String | 50 | 否 | 注册时用用户名做昵称 个人中心可以修改 |
| birthday | 年龄 | Date |  | 是 |  |
| gender | 性别 | int | 10 | 是 | 枚举实现 |
| locked | 是否禁止 | boolean |  | 否 | 禁止后不能登录，true-禁止，false-开放，默认值为false |
| create\_time |  | datatime |  | 否 |  |
| update\_time |  | datatime |  | 否 |  |

## 栏目表 cms\_channel

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段说明 | 字段类型 | 字段长度 | 为空 | 备注 |
| id | 栏目id | Integer | 10 | 否 | 唯一主键，自增长 |
| name | 栏目名称 | String | 50 | 否 |  |
| description | 栏目说明 | String | 500 | 是 |  |
| icon | 栏目图标 | String | 30 | 是 | 使用**fontawesome** 字体图标 |

## 分类表 cms\_category

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段说明 | 字段类型 | 字段长度 | 默认值 | 为空 | 备注 |
| id | 分类id | Integer | 10 | 1 | 否 | 唯一主键，自增长 |
| name | 分类名称 | String | 50 |  | 否 |  |
| channel\_id | 栏目id | int | 10 |  | 否 |  |

## 文章内容表cms\_article

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段说明 | 字段类型 | 长度 | 默认值 | 为空 | 备注 |
| id | 文章id | Integer | 10 | 1 | 否 | 唯一主键 |
| title | 标题 | String | 100 |  | 否 |  |
| content | 文章内容 | longtext |  |  | 否 |  |
| picture | 图片 | String | 50 |  | 是 | 如果没上传，则在文章内容自动提取，内容也没有图片，则为空 |
| channel\_id | 栏目id | int | 10 |  | 否 |  |
| category\_id | 类型id | int | 10 |  | 否 |  |
| user\_id | 发布人id | int | 10 |  | 否 |  |
| hits | 点击量 | int | 20 | 0 |  |  |
| hot | 是否上热门 | boolean | 10 | 0 | 否 | true:上 false:不上 |
| status | 是否审核通过 | int | 10 | 0 | 否 | -1:不通过，1:通过 |
| deleted | 是否有效 | int | 10 | 1 | 否 | **0:失效1:有效 使用逻辑删除** |
| created |  | datatime |  |  | 否 |  |
| updated |  | datatime |  |  | 否 |  |

## 广告表 cms\_slide

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段说明 | 字段类型 | 字段长度 | 默认值 | 为空 | 备注 |
| id | 图片id | Integer | 10 | 1 | 否 | 唯一主键 |
| title | 标题 | String | 50 |  | 是 |  |
| picture | 幻灯片图片 | String | 100 |  | 是 |  |
| url | 跳转地址 | String | 100 |  | 否 | 站内写文章最终页地址 |

## 配置表 cms\_settings

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段说明 | 字段类型 | 长度 | 默认值 | 为空 | 备注 |
| site\_domain | 网站域名 | String | 50 | localhost | 否 | 没有主键 |
| site\_name | 网站名称 | String | 50 |  | 否 |  |
| article\_list\_size | 文章列表页显示数量 | int | 20 | 20 | 否 |  |
| slide\_size | 首页幻灯片数量 | int | 10 | 5 | 是 |  |
| admin\_username | 管理员账号 | String | 16 | admin | 否 |  |
| admin\_password | 管理员密码 | String | 32 | 888888 | 否 | MD5值 |

# 整个项目设计

整个项目分前台和后台,后台又分两个:1,普通注册用户的后台,2管理员后台

## 前台

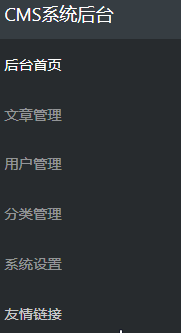
前台主要是系统的显示功能,包括按照栏目\分类显示文章,并包含登录,注册等功能. 将来可以增加文章的评论,友情链接,文章收藏等功能.

## 普通注册个人后台

维护个人信息(密码,头像,昵称),发表文章,维护个人收藏,



## 管理员后台



必须完成的功能,文章管理(审核文章),用户管理(普通账户停用,查询等功能)

# 开发计划

## 系统搭建

### 表的创建

### 实体类,映射文件及相关接口

### 三层

## 管理员后台

### 用户管理

功能:

1. 查看系统下所有注册的用户信息
2. 禁用(用不不能)或启用(只有启用的用户才能登陆)用户

开发思路:

从后向前即: dao-->service-->controller-->jsp(需要使用bootstrap统一样式和布局)

## Dao

|  |
| --- |
| package com.lisi.cms.dao;  import java.util.List;  import java.util.Map;  import org.apache.ibatis.annotations.Param;  import com.lisi.cms.domain.User;  /\*\*  \*  \* @ClassName: UserMapper  \* @Description: 维护用户基本信息  \* @author: charles  \* @date: 2019年5月20日 下午3:18:52  \*/  public interface UserMapper {  /\*\*  \*  \* @Title: selectListByName  \* @Description: \* 按照名称模糊查询  \* @param name  \* @return  \* @return: List<Map>  \*/  List<Map> selectListByName(@Param("username") String username);  /\*\*  \*  \* @Title: updateByPrimaryKeySelective  \* @Description: 如果User 的locked是1:则为禁用. 0:表示启用  \* @param record  \* @return  \* @return: int  \*/  int updateByPrimaryKeySelective(User record);  int deleteByPrimaryKey(Integer id);  /\*\*  \*  \* @Title: insert  \* @Description: 增加用户或注册用户  \* @param record  \* @return  \* @return: int  \*/  int insert(User record);  /\*\*  \*  \* @Title: getUser  \* @Description: 登录  \* @param name  \* @return  \* @return: User  \*/  User getUser(String name);  int insertSelective(User record);  User selectByPrimaryKey(Integer id);  int updateByPrimaryKey(User record);  } |

## Usermapper.xml

|  |
| --- |
| 省略.... |

## Service及实现类

|  |
| --- |
| package com.lisi.cms.service.impl;  import java.util.List;  import java.util.Map;  import javax.annotation.Resource;  import org.springframework.stereotype.Service;  import com.lisi.cms.dao.UserMapper;  import com.lisi.cms.domain.User;  import com.lisi.cms.service.UserService;  @Service  public class UserServiceImpl implements UserService {  @Resource  private UserMapper userMapper;  @Override  public List<Map> selectListByName(String name) {  return userMapper.selectListByName(name);  }  @Override  public int updateByPrimaryKeySelective(User record) {  return userMapper.updateByPrimaryKey(record);  }  @Override  public User selectByPrimaryKey(Integer id) {  return userMapper.selectByPrimaryKey(id);  }  @Override  public User getUser(String name) {  return userMapper.getUser(name);  }  } |

### 测试

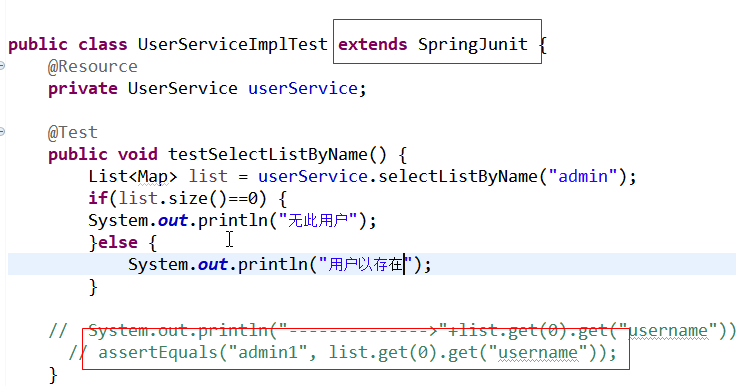
使用springjunt4的整合测试

## 先开发测试父类,启动spring

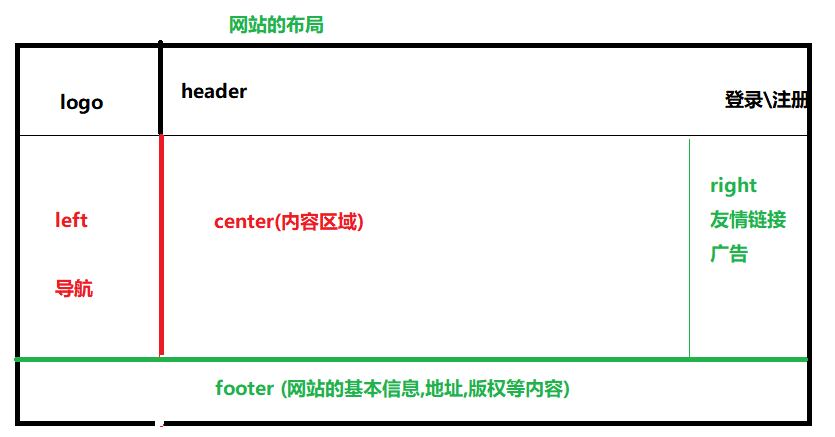


|  |
| --- |
| /\*\*  \* 测试类启动spring容器  \* **@ClassName**: SpringJunit  \* **@Description**: **TODO**  \* **@author**: charles  \* **@date**: 2019年5月21日 上午8:58:34  \*/  @RunWith(SpringJUnit4ClassRunner.**class**)  @ContextConfiguration(locations="classpath:spring-beans.xml")  **public** **class** SpringJunit {  } |

## 所有的测试类继承该父类



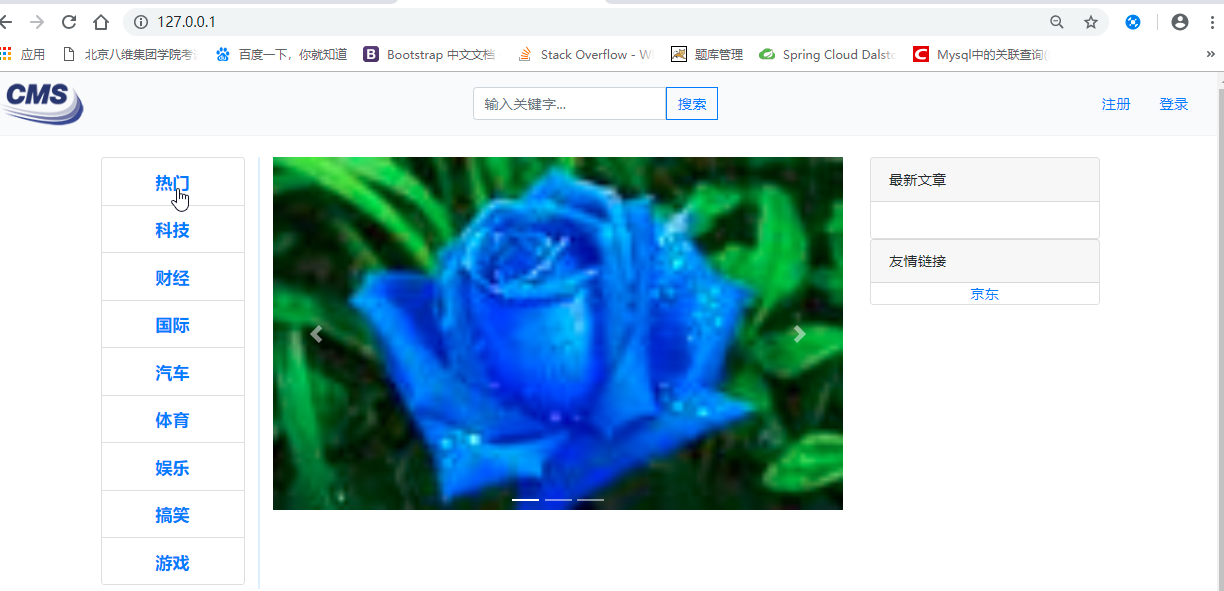
## 后台布局



本系统使用bootstrap 前端框架布局

## 本节段要实现的功能

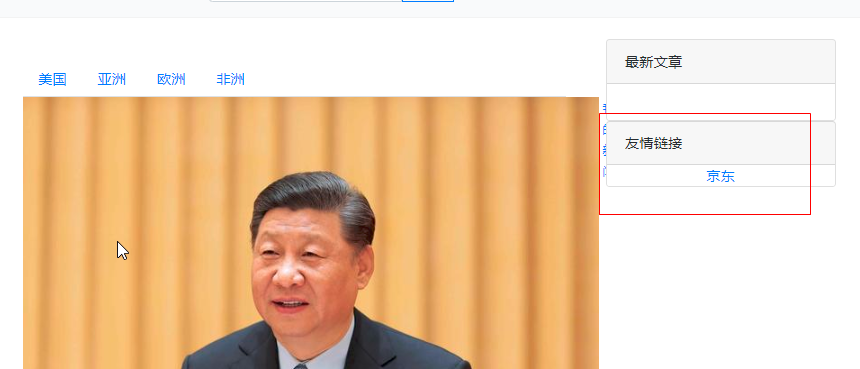
### 前台栏目展示



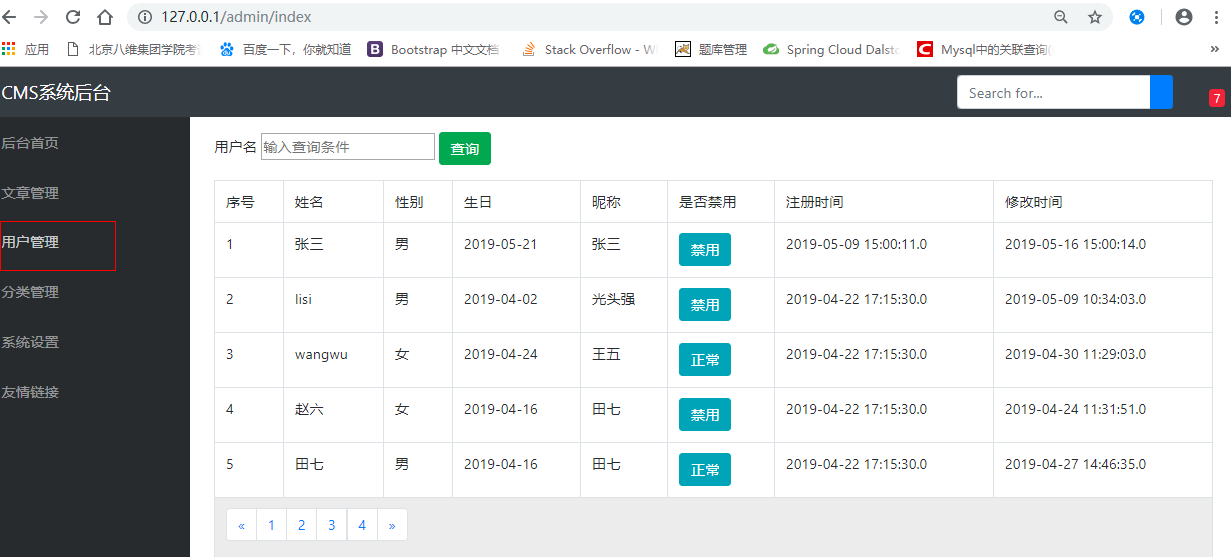
### 前台分类展示



### 前台友情链接展示



### 管理员后台用户管理



### 管理后台友情链接



### 管理员后文章审核管理

### 个人后台文章发布