WEB应用设计

一.交互设计

通过导航给予提示信息，减轻用户的认知压力





易于识别的功能标识





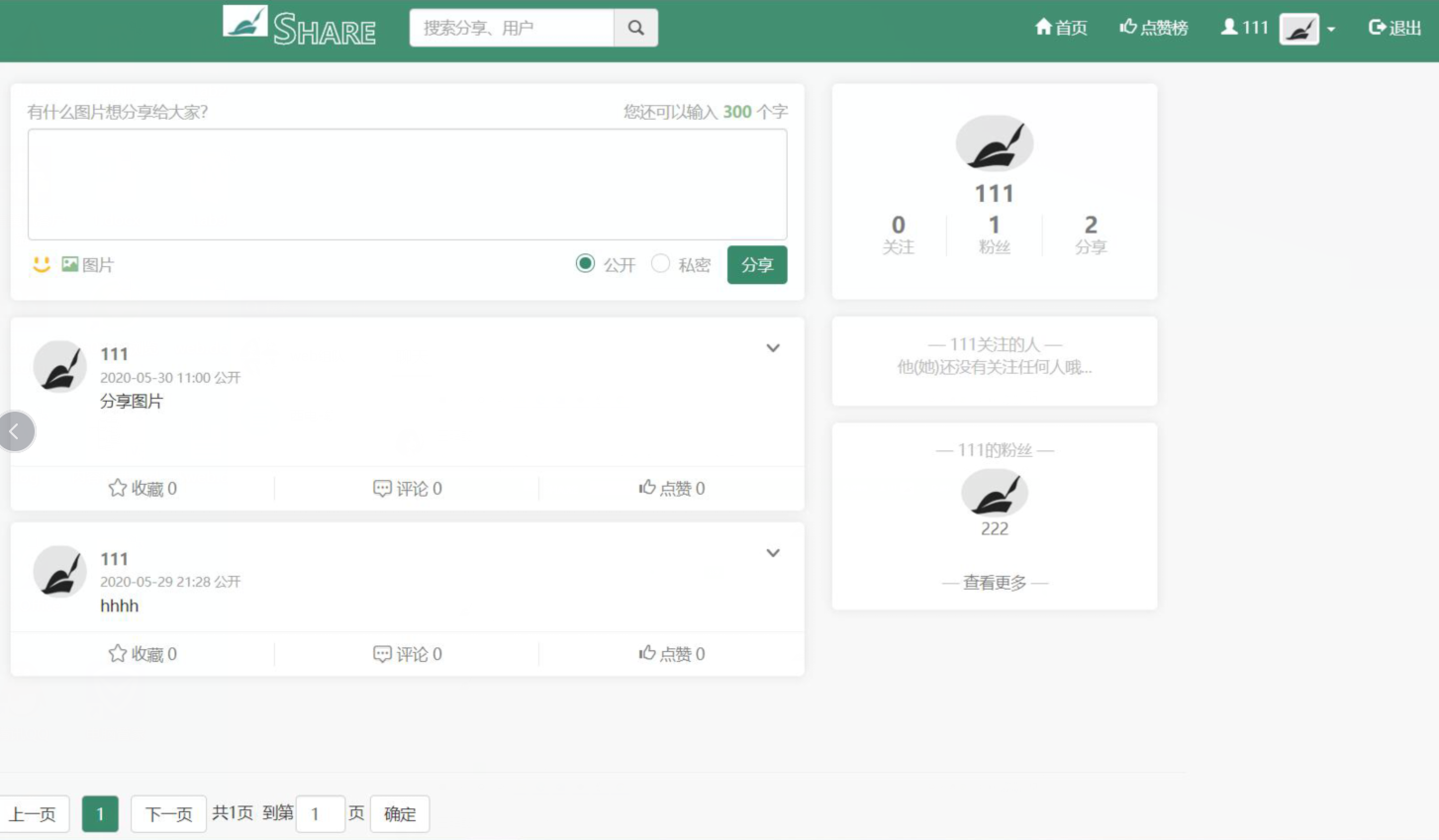
二.展示设计

页面布局

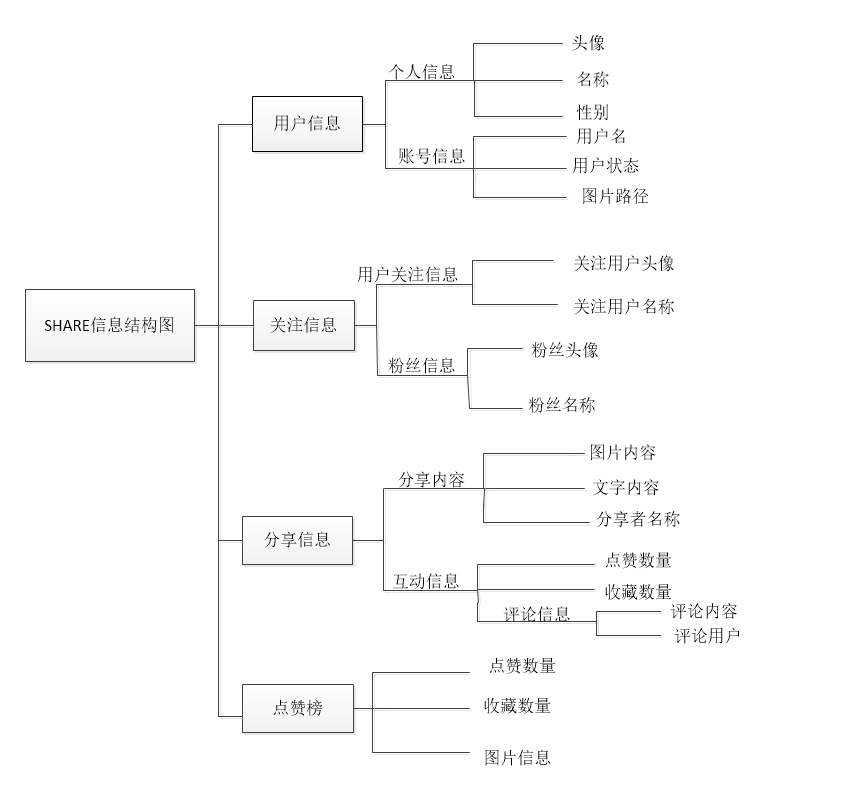
采用上下框架型布局

上方为功能模块，下方为主要内容





三.内容设计

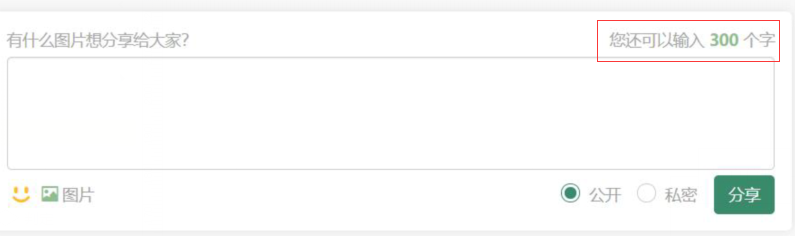


四.功能设计

尽可能减少⽤户对记忆的依赖



用户交互后可以获得反馈



显示哪些项目是可点击可交互的



提供有用和支持性的出错提示





WEB应用构建与部署

1. 系统架构

采用MVC+三层架构的方式进行程序的开发

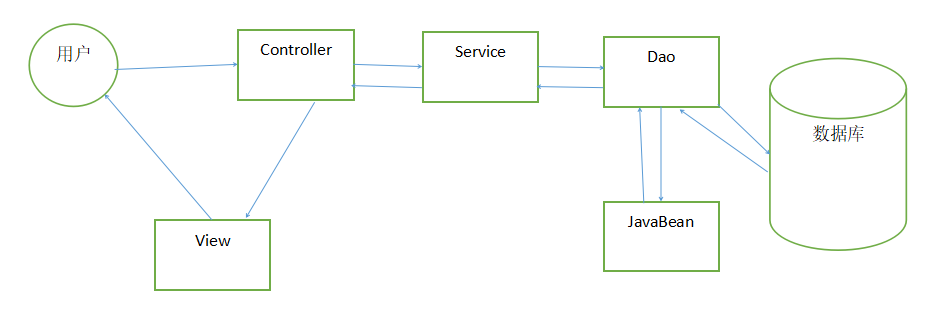
Controller:负责接收用户请求，获取请求参数，调用业务方法，并将数据转发到视图

Service:负责处理业务逻辑，调用Dao操作数据库

Dao:负责和数据库打交道

JavaBean:负责封装数据

View:视图，负责展示数据



二.用到的后台框架

①Spring

主要用到了IOC和AOP两大功能：

IOC:控制反转，IOC容器负责创建对象，我们的代码面向接口，只需要直接调用，而不需要我们自己来创建对象。

AOP:面向切面，利用动态代理，抽取出一些重复的代码。

②Hibernate

一个持久层的ORM框架，方便访问数据库

③SpringMVC

Spring提供的MVC框架，主要做了封装请求参数，转发数据到视图，以及将返回的数据转化为JSON等便利的功能。

三.数据库

1.表

分享表，类似一条微博，包括文字和图片

share(id,content,datetime,status)--status 0代表可见 1代表不可见

分享图片表

shareImg(id,shareId,imgPath)

点赞表，对分享点赞

like(userId,shareId)

评论表，评论某个分享

1.第一条评论的parentId为0

2.可以回复评论，被回复的评论设置为回复的parentId

3.parentId为0的评论放在第一层

4.parentId不为0的评论，找到自己的祖先，即parentId为0的节点，全部放置在第二层，按时间排序

comment(id,userId,shareId,parentId,content,datetime)

收藏表，收藏其他用户的分享

collect(userId,shareId)

关注表，可以关注其他用户

follow(userId,fansId);

用户表

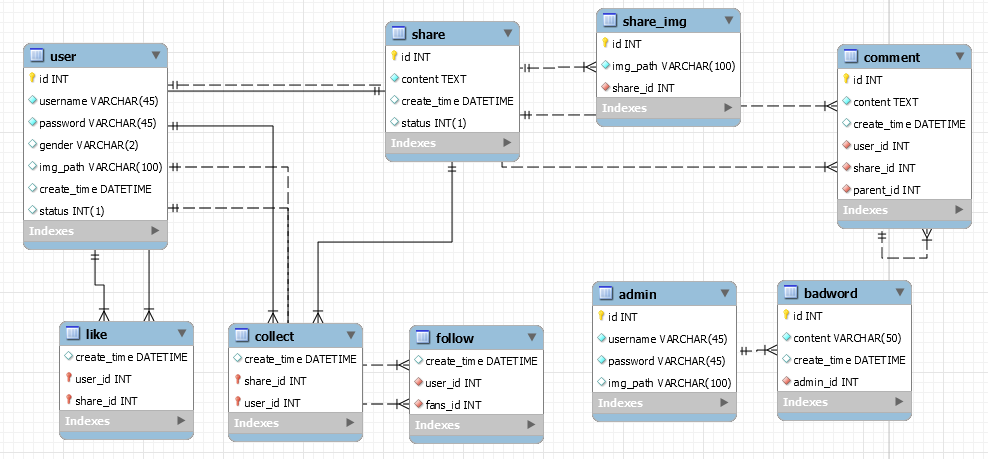
user(id,username,password,gender,imgPath,status)--status 0可用，1禁用

管理员表

admin(id,username,password,imgPath); //导入mysql中，自己把admin改为admin\_ (不然会报错)

屏蔽词语表

badword(id,adminId,content);



四、页面

用户部分

注册、登录页面（可以选择弹出框的形式）

发布分享页面（textarea+最新分享列表，每条分享包括用户头像、内容、点赞数量、评论、收藏按钮等内容）

个人主页，包括个人基本信息和个人分享列表，在个人主页可关注该用户

--查看个人收藏的分享

--查看个人关注的用户

--查看自己点赞过的分享

--查看自己评论过的分享

管理员部分

管理员登录页面

用户列表页面，可禁止或允许用户的分享活动

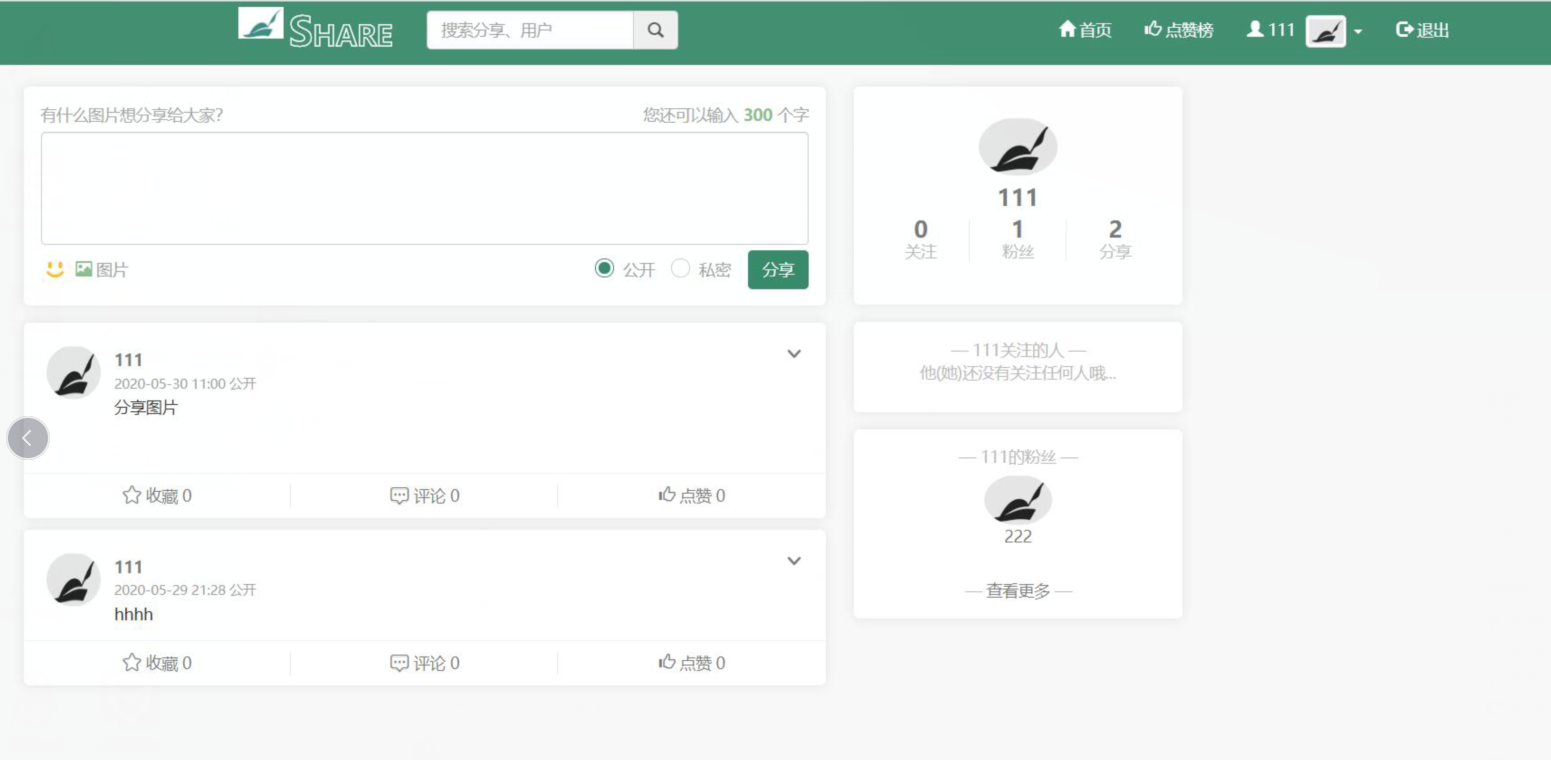
屏蔽关键字列表页面，可对关键字进行增删查改

页面展示

注册/登录页面



用户登录后页面



管理员页面

