**华南理工大学海南同乡会网站2.0概要设计说明书**

**版本 1.0**

|  |  |  |  |
| --- | --- | --- | --- |
| 作者： | 开发组全体成员 | 日期： | 2015/11/9 |
| 审批： |  | 日期： |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 变更说明 | 作者 |
| 2015/11/9 | 1.0 | 创建 | 开发组全体成员 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 简介

### 内容

本文档主要内容为华南理工大学海南同乡会网站2.0架构的详细说明，以及web层、业务领域层和数据持久层的解决方案说明，为后续开发提供技术解决依据。

### 适用对象

适用于架构和开发人员以及测试人员。

### 参考文档

《需求规格说明书》

《》

## **系统整体架构**

## **概述：**

前台（Web层）使用jsp+css+jQuery。

后台（业务逻辑层和持久层）使用SSH（控制器、业务逻辑操作、持久化操作）。

前后台使用json进行数据交互。

数据库使用mysql，可选择利用主从复制实现读写分离。

使用nginx做反向代理服务器,实现负载均衡，tomcat做逻辑服务器，apache做图片服务器。

生产环境在Linux系统下，开发环境可使用Windows/Linux+tomcat（localhost）。

## **总体服务器架构：**

**代理服务器**：对所有访问网站应用的请求（包括对逻辑服务器和图片服务器的请求）进行转发和负载均衡配置，在该网站应用中，该代理服务器采用的是反向代理，进行了反向代理之后，隐藏了后端的服务器群，提高了一些安全性。另外通过负载均衡，降低了服务器的负担，使网站整体更加稳定。

**图片服务器**：用于处理网站用户上传的图片以及图片请求，同乡会网站2.0的图片服务器依托新浪云SAE免费服务器，使用php开发图片服务器端处理应用，处理图片文件的接收与存储，并与逻辑服务器进行通信，确保图片信息在数据库中的一致性，目前计划开发，配置3~5台图片服务器,命名方式为 scuthnimage1.sinaapp.com, scuthnimage2.sinaapp.com ...

**逻辑服务器**: 主要用于进行网站主要业务逻辑应用的部署与架设,网站主体应用采用J2EE的SSH( Struts2 + Spring + Hibernate )框架进行开发，因此逻辑服务器采用作为servelet容器的Tomcat,该服务器主要进行业务请求的处理，以及与图片服务器和数据库服务器之间的的通信.计划配置一台逻辑服务器,但是在业务量急剧增加的情况下,可以按情况增加服务器数量.

**数据库服务器**：用于接收处理数据请求，并对数据实体进行相关操作以及管理，网站采用数据库服务器为Mysql,开放的端口为127.0.0.1(localhost):3306.

**数据实体**:网站应用的相关业务数据以及服务器数据,交由mysql进行管理,具体表现为数据库的/data下的数据文件.

**物理主机:**

**新浪云SAE**：新浪云主机（Sina Elastic Cloud 简称SEC）是潜心打造的新一代简单、可靠  
的IaaS服务，为您提供基于安全可靠的基础网络的高性能易扩展的计算服务，选择该服务器作为图片服务器的原因是,该云主机在一定程度上可以免费使用,并且每个用户可以开启50G的免费磁盘空间,虽然限制了本地的IO操作,但是可以通过SAE官方提供的storage API进行文件存储管理，并且服务其提供免费二级域名，可以直接通过该域名访问服务器内的静态资源，在当前服务器磁盘价格较高的情况下，可以牺牲一些灵活性，换取免费的空间。

**阿里云ECS：**云服务器（Elastic Compute Service，简称ECS）是一种简单高效、处理能力可弹性伸缩的计算服务，帮助您快速构建更稳定、安全的应用，提升运维效率，降低IT成本，使您更专注于核心业务创新。

## **应用架构：**

1. **主应用架构(逻辑服务器端)**

**主应用采用使用SSH框架实现的MVC架构.**

View：view层主要功能为,作为后台代理与用户进行交互，收集用户输入数据交由后台处理，并呈现数据给用户。

view为采用Jsp进行整合的页面，前端采用的框架与技术主要为jsp+css+jQuery+bootstrap

**前端与后台有两种请求响应方式:**

1. **第一种**为ajax请求与响应,前端使用ajax方法向后台发送请求,后台进行处理之后返回json数据,再由前端进行页面的局部更新.请求过程如下图：

前端页面

后台

前端页面

后台

前端页面

后台

1. **第二种**为页面请求与响应, 即前端请求( 一般是通过点击一个页面请求链接,或者表单提交等等 )一个新的html页面，后台将发送来的请求作处理，将处理获取的数据（如果需要的话）在servelet容器内jsp向html转化时进行整合，并将得到的html页面发送至前端，前端获取html页面并做页面替换刷新.请求过程如下图：

前端项目编码结构：

前端文档目录主要涉及4个文件夹，css、js、images、jsp

css目录结构如下：



其中base存放初始设置的css文件，common分别存放前台跟管理员页面的统一设置的css文件，frame存放一些框架的css文件，page里面分别存放前台跟管理员每个页面具体设置的css文件。

Js目录结构如下：



其中base存放一些封装的函数，如重新封装的ajax函数等；common分别存放前台跟管理员页面公共函数的js文件，frame存放一些框架的js文件，page里面分别存放前台跟管理员每个页面具体设置的js文件。

Images目录如下：

D:\教学资料\实训\j2ee项目\work doc\requirement\manage\前端目录结构文档\image.png

只有一个文件夹，里面存放页面所用的所有图片资源文件

Jsp目录结构如下：



Home.jsp为项目的入口文件，manage为后台管理员的jsp文件，market为前台浏览的jsp文件。

**Controller**：controller层的主要功能为对前端发送而来的请求数据进行转发，转交给不同Model中的service组件进行业务逻辑的处理，并对service组件所返回的数据进行封装(封装成为json或者jsp )并返回给前端，该层前端有多个拦截器和过滤器，用于进行权限检查，数据校验 以及 XSS攻击过滤 和 SQL注入攻击过滤 等等。

编码实现：

命名规范：

**Model**：model层主要用于进行业务逻辑的处理以及数据持久化，为了结构和处理流程的清晰，将model分为Service（服务组件）和Dao（data access object 数据访问对象）两部分,前者职责为进行业务逻辑处理，后者职责为数据持久化.

编码实现：

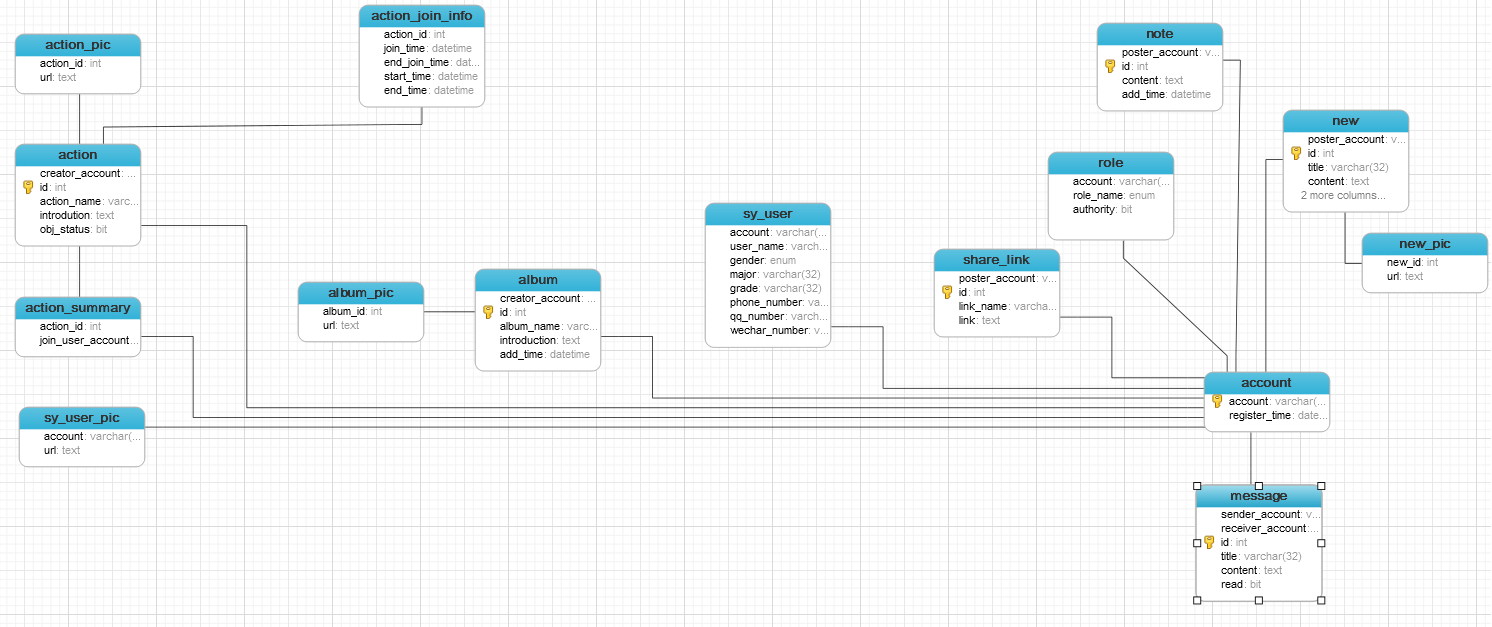
命名规范：

1. **图片服务器端应用架构**

图片服务器端应用使用php语言，依托php原生API和SAE提供的相关API进行开发，图片服务器端实现的功能主要有两个：

1. 获取用户上传的图片，并将其存储至服务器中
2. 与逻辑服务器进行通信，将存储后的图片url告知于逻辑服务器，逻辑服务器更新数据库，以保持数据一致性。
3. **数据库设计**

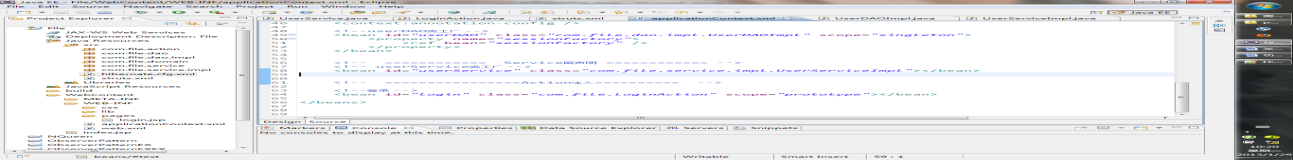
3.1 数据库设计的概念模型

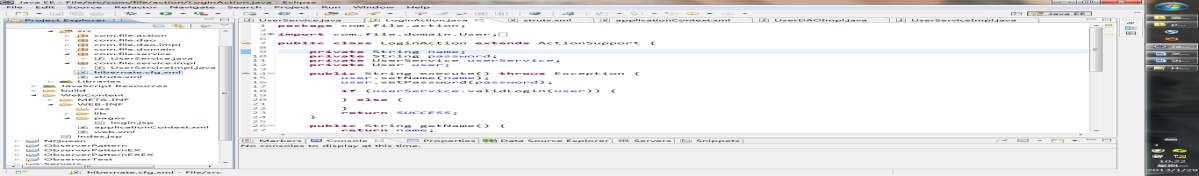


## **接口设计**

所有service和dao组件采用面向接口编程的方式，如上图例，使用spring IOC控制反转原则将实现类注入，使用接口定义对象，最大程度减少耦合，增大可维护性。

例如：在代码中申明都用接口，但是在applicationContext.xml中注入相应的实现类





UserService.java是接口，而UserServiceImpl是它的实现类。

在代码里面声明是用UserService接口，而在配置文件中是用UserServiceImpl实现类，这样就是说用spring容器把实现类通过setUserService（）函数注入到代码中了，这样达到松耦合的目的。

**命名规范（注：所有包名小写，所有类名采用驼峰写法）：**

1. **资源层命名规范：**

**包名: com.scuthnweb.dao**

**接口包名: com.scuthnweb.dao**

**实现包名: com.scuthnweb.dao.impl**

**接口名: NoteDao (Note 表名加上Dao后缀)**

**实现类名：NoteDaoImpl （Note表名加上DaoImpl后缀）**

1. **服务组件命名规范：**

**包名: com.scuthnweb.service**

**接口包名: com.scuthnweb.service**

**实现包名: com.scuthnweb.service.impl**

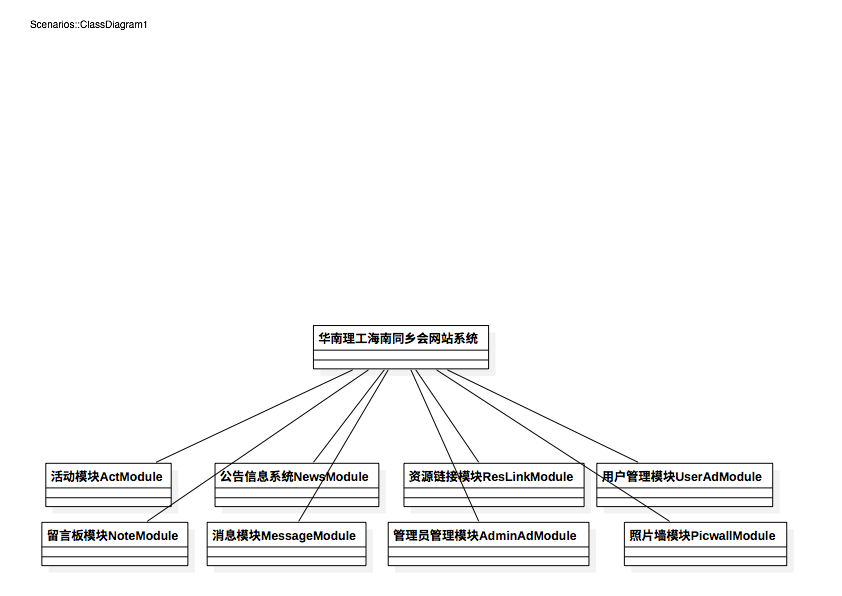
**接口名: NoteModule (NoteModule 模块名)**

**实现类名：NoteModuleImpl （Note模块名加上Impl后缀）**

1. **资源（Dao）层类与接口设计**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 类名称 |  | | | 所属包名称 | | |  | |
| 类静态变量定义 | | | | | | | | |
| 名称 | | | 类型 | | | 描述 | | 范围 |
|  | | |  | | |  | |  |
| 类变量定义 | | | | | | | | |
| 名称 | | | 类型 | | | 描述 | | 范围 |
|  | | |  | | |  | |  |
| 方法 | | | | | | | | |
| 名称 | | 参数 | | | 返回值 | | 简单描述 | 范围 |
|  | |  | | |  | |  |  |

1. **服务组件（Service）类与接口设计**



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 类名称 |  | | | 所属包名称 | | |  | |
| 类静态变量定义 | | | | | | | | |
| 名称 | | | 类型 | | | 描述 | | 范围 |
|  | | |  | | |  | |  |
| 类变量定义 | | | | | | | | |
| 名称 | | | 类型 | | | 描述 | | 范围 |
|  | | |  | | |  | |  |
| 方法 | | | | | | | | |
| 名称 | | 参数 | | | 返回值 | | 简单描述 | 范围 |
|  | |  | | |  | |  |  |

1. **图片服务器端应用接口设计**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 类名称 |  | | | 所属包名称 | | |  | |
| 类静态变量定义 | | | | | | | | |
| 名称 | | | 类型 | | | 描述 | | 范围 |
|  | | |  | | |  | |  |
| 类变量定义 | | | | | | | | |
| 名称 | | | 类型 | | | 描述 | | 范围 |
|  | | |  | | |  | |  |
| 方法 | | | | | | | | |
| 名称 | | 参数 | | | 返回值 | | 简单描述 | 范围 |
|  | |  | | |  | |  |  |