|  |  |
| --- | --- |
| 项目编号 |  |
| 文档编号 |  |
| 密级 |  |

鲲鹏素拓社团门户网站概要设计

V1.0

桂林理工大学软件工程

评审日期：2020年 1 月1日

目录

[**1. 导言** 6](#_Toc52290645)

[**1.1 编写目的** 6](#_Toc52290646)

[**1.2 项目范围** 6](#_Toc52290647)

[**1.3 参考资料** 6](#_Toc52290648)

[**1.4 版本更新信息** 7](#_Toc52290649)

[**2. 项目设计原则简介** 7](#_Toc52290650)

[**2.1 实用性** 7](#_Toc52290651)

[**2.2 先进性** 7](#_Toc52290652)

[**2.3 以用户为中心的处理** 7](#_Toc52290653)

[**2.4 使用便捷** 8](#_Toc52290654)

[**2.5 灵活、易维护** 8](#_Toc52290655)

[**2.6 安全可靠** 8](#_Toc52290656)

[**2.7 单一职责原则** 8](#_Toc52290657)

[**3. 体系结构设计** 9](#_Toc52290658)

[**3.1 表现层** 11](#_Toc52290659)

[**3.2 控制层** 14](#_Toc52290660)

[**3.3 业务逻辑层** 15](#_Toc52290661)

[**3.4 数据持久层** 16](#_Toc52290662)

[**3.5 域模型层** 18](#_Toc52290663)

[**3.6 接口设计** 19](#_Toc52290664)

[3.6.1 内部接口 19](#_Toc52290665)

[**4. 功能模块设计** 21](#_Toc52290666)

[**4.1 功能模块设计总述** 21](#_Toc52290667)

[**4.2 客户端子系统模块** 24](#_Toc52290668)

[4.2.1 模块 SCM1：主页 24](#_Toc52290669)

[4.2.2 模块 SCM2：社团信息 24](#_Toc52290670)

[4.2.3 模块 SCM3：登录模块 24](#_Toc52290671)

[4.2.4 模块 SCM4：注册模块 25](#_Toc52290672)

[4.2.5 模块 SCM5：历史记录 25](#_Toc52290673)

[4.2.6 模块 SCM6：活动信息 26](#_Toc52290674)

[4.2.7 模块SCM7：公告 26](#_Toc52290675)

[4.2.8 模块SCM8：社团加入 27](#_Toc52290676)

[**4.3 管理端子系统模块** 27](#_Toc52290677)

[4.3.1 模块DAM1：用户管理 27](#_Toc52290678)

[4.3.2 模块DAM2：公告发布 28](#_Toc52290679)

[4.3.3 模块DAM3：后台数据查询 28](#_Toc52290680)

[**5．数据库设计** 29](#_Toc52290681)

[**5.1 数据库种类及特点** 29](#_Toc52290682)

[**5.2 数据库逻辑结构** 29](#_Toc52290683)

[**5.3 物理结构设计** 30](#_Toc52290684)

[5.3.1 Customer表 30](#_Toc52290685)

[5.3.2 Auditor表 31](#_Toc52290686)

[5.3.3 Buyer表 31](#_Toc52290687)

[5.3.4 Textbook表 32](#_Toc52290688)

[5.3.5 Stock表 32](#_Toc52290689)

[5.3.6 Out\_of\_Stock表 33](#_Toc52290690)

[5.3.7 In\_Storage表 33](#_Toc52290691)

[5.3.8 Textbook\_Lack表 34](#_Toc52290692)

[5.3.9 Textbook\_Purchase表 34](#_Toc52290693)

[5.3.10 Textbook\_Order表 36](#_Toc52290694)

[**6. 界面设计** 36](#_Toc52290695)

[**6.1 首页设计** 36](#_Toc52290696)

[**6.2 登录注册** 37](#_Toc52290697)

# **1. 导言**

## **1.1 编写目的**

该文档描述了鲲鹏素拓社团门户网站的功能和性能的要求，将作为对该项目在概要设计阶段的设计输入。

本文档的预期读者包括：

* 设计开发人员
* 项目管理人员
* 测试人员
* 用户

## **1.2 项目范围**

该文档的目的是解决整个项目系统中“做什么”的问题。对于开发技术在这里并没有涉及，而主要是通过建立模型的方式来描述用户的需求为，客户、用户、开发方等不同参与方提供一个交流的平台。

## **1.3 参考资料**

1. 《软件工程案例教程 第 2 版》 韩万江等 机械工业出版社
2. 《软件项目管理案例教程 第 3 版》韩万江等，机械工业出版社

## **1.4 版本更新信息**

本文档的更新记录如表 1-1 所示。

表 1-1 版本更新信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修改编号 | 修改日期 | 修改后版本 | 修改位置 | 修改内容概述 |
| 001 | 2020.9.29 | 1.0 | 目录 | 初始发布版本 |

# **2. 项目设计原则简介**

社团门户网站使用最广泛用于开发web端应用的框架，配合社团的独特性，方便学生了解社团，加入社团。提高了社团人员的办事效率。在整个系统设计的过程中遵循以下的设计原则：

## **2.1 实用性**

实用性是系统的主要设计原则，系统设计必须最大可能地满足用户的需求，做到操作方便、界面友好、可即时更新，能适应不同层次用户的需求。

## **2.2 先进性**

信息技术发展迅速，系统设计尽可能采用先进的技术标准和技术方法。

## **2.3 以用户为中心的处理**

个性化服务充分体现了这一点，根据用户当前展业重点，配置页面功能布局及展现内容，贴合用户操作。

## **2.4 使用便捷**

系统要有设计良好的人机交互界面，即使系统的操作界面简单易用，又能具有较强的适用性，满足不同计算机使用水平的用户使用。

## **2.5 灵活、易维护**

本系统采用开放的体系架构，基于开放源代码的技术框架和数据库系统，使用高效率的开源和免费开发工具，具备完整的文档说明。在维护方面，主要考虑两个层面，一是对于开发人员来讲，系统编码容易调整，可适应需求的变化和调整；二是对于系统管理维护人员来说，能够对系统进行便捷的维护和管理。

## **2.6 安全可靠**

选择安全可靠的软硬件运行平台，并在系统设计和实现的时候关注系统的安全控制和执行效率，提供相应的安全防护功能，保证系统具有较高的安全性和可靠性。安全性方面，要考虑系统的安全、数据管理的安全、网络安全。保证用户权限、数据安全和系统的稳定性。

## **2.7 单一职责原则**

我们系统在面向对象设计部分采取单一职责原则，其核心思想为：一个类，最好只做一件事，只有一个引起它的变化。单一职责原则可以看做是低耦合、高内聚在面向对象原则上的引申，将职责定义为引起变化的原因，以提高内聚性来减少引起变化的原因。从而最终提高我们系统的可修改性和可维护性。

本概要设计涵盖了体系结构设计、模块设计、数据库设计、界面设计等。

# **3. 体系结构设计**

鲲鹏素拓社团门户网站本着软件开发的设计原则，采用浏览器/服务器（B/S）的体系结构。为了满足系统响应快速、便于操作、易于维护的要求，在软件架构上，采用五层体系结构：表现层、控制层、业务逻辑层、数据持久层和域模型层；在设计实现上，我们采用 MVC 的设计模式: Model 模型层、View 视图层、Controller控制层；在体系架构上，SPM 网络课堂系统选择用 SSM 架构。

SSM在此处是指Spring MVC + Spring Boot + Mybatis 的一个集成框架，是目前较流行的一种Web应用程序开源框架。鲲鹏素拓社团门户网站集成的 SSM 框架的系统从分为五层：表现层、控制层、业务逻辑层、数据持久层和域模型层，以帮助开发人员在短期内搭建结构清晰、可复用性好、维护方便的 Web 应用程序。其中使用 Spring Boot 作为系统的整体基础架构，负责 MVC 的分离。Spring MVC 框架作为模型部分，控制业务跳转。利用Mybatis框架对持久层提供支持。

具体做法是：用面向对象的分析方法根据需求提出一些模型，将这些模型实现为基本的 Java 对象，然后编写基本的 DAO(Data Access Objects)接口，并给出Mybatis 的 DAO 实现，采用 Mybatis 架构实现的 DAO 类来实现 Java 类与数据库之间的转换和访问。

网络结构图如图3-1所示，用户（用户和信息管理员）可以通过网络登录到网站，进行相应的操作，应用服务器接收到相应操作反映之后对数据库进行相应操作，并返回进行展示。

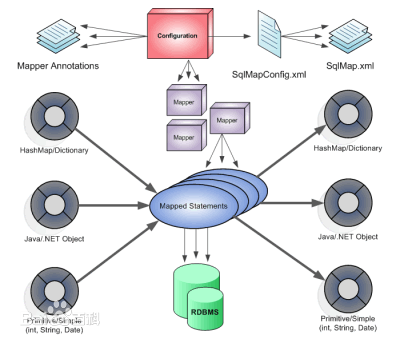


图3-1 mybatis底层结构图

系统技术架构统如图3-2所示。

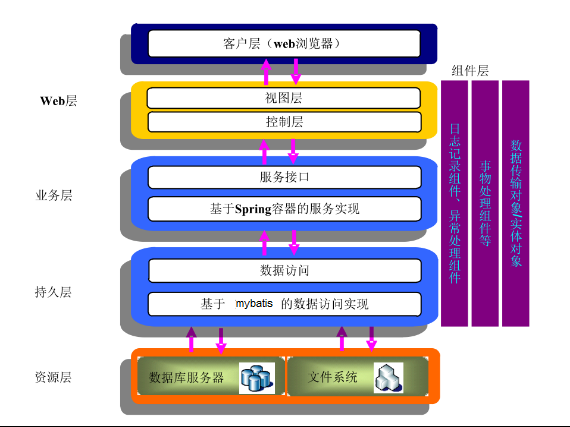


图 3-2系统分层结构图

## **3.1 表现层**

结合用户身份判定，用于展示系统的业务信息以及接受用户输入信息。将来自用户的信息发送到对应的业务层进行处理，并接受后台处理的结果，并结合用户身份将其返回到前端进行展示，实现系统与用户的动态交互。

表现层功能包括：

\* 根据用户身份信息进行识别登录，显示不同界面。

\* 根据用户身份信息进行识别实用系统，进行不同的数据和逻辑交互，满足用户需求 。

\* 采用 Spring MVC 标签库以及 JSTL 标签库开发 JSP 页面。

\* 通过页面控件，完成统一风格的页面部署，形成较好的页面风格。

\* 采用多种展现形式，html 页面，excel 等，图片，PDF，视频播放插件等等。

\* 对前端提交数据进行合规性校验，如登入时账号密码长度，类型，格式等等。

\* 将页面请求发送到验证层，并接受验证层返回，将结果在前端展示。

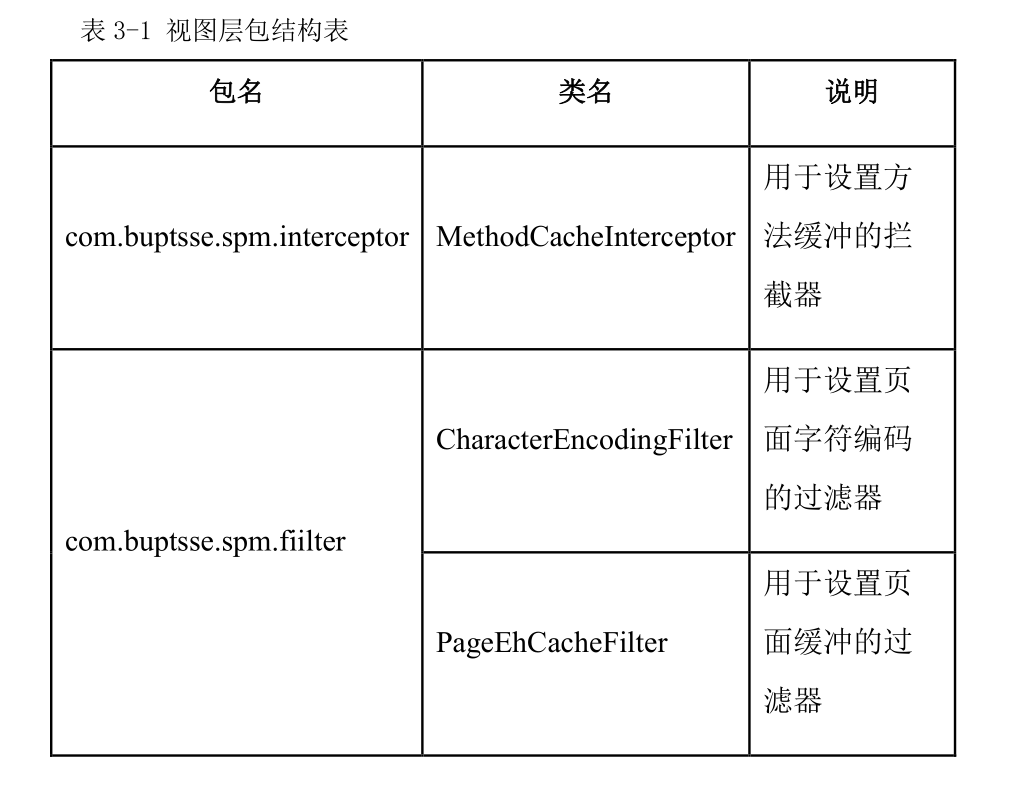
\* 提供过滤器，进行请求预处理。

\* 提供文件上传功能

表现层主要由下面组件构成：

1. 分发器与拦截器组件：

该组件统一处理分发来自页面的请求，通常将所有的请求分发情况都配置到 Spring MVC 的配置文件中。分发器与拦截器包结构见表 3-1。



2. 网页视图组件：

该组件根据业务逻辑层的处理数据进行页面的展示。网页视图部分样例表

表 3-2 网页视图部分样例表

|  |  |  |
| --- | --- | --- |
| 文件名称 | html | 说明 |
| 社团信息 | organizationInformation.html | 显示社团信息 |
| 用户信息 | userInformation.html | 显示用户信息 |
| 活动信息 | activityInformation.html | 显示活动信息 |
| 历史记录 | history.html | 显示过去的活动记录、公告等 |

## **3.2 控制层**

根据不同的请求触发点，接收从表现层传输的用户请求信息，并将封装好的实体对象发送到对应的业务处理单元，同时接收业务逻辑层处理结果，指定相应的表现层 jsp 页面展现需求数据，实现页面跳转与信息现实。

控制层功能包括：

\* 接收并处理从表现层传入的各种输入以及输出各种异常提示信息或

处理结果信息。

\* 对于输入的数据进行数据校验，过滤非法数据。

\* 向业务控制层发送处理请求。

\* 接收业务逻辑层请求处理结果。

\* 返回状态符，实现页面跳转。

控制层主要用下面组件构成：

1.控制器组件：执行由分发器发过来的请求。该模块还继承自 Spring MVC 的ActionSupport 实现了一些扩展，这个扩展提供了一些公用操作，比如获取

Request、Session 对象等操作。所有业务模块的 Action 都可以继承该类。控制层包部分结构见表 3-3。

表 3-3 控制层包部分结构

|  |  |  |
| --- | --- | --- |
| 包名 | 类名 | 说明 |
| com.KunpengOrganization.action | BasicInfoAction | 封装 Spring MVC 2 的ActionSupport 基类，实现基本信息的控制跳转 |
|  | LoginAction | 封装 Spring MVC 2 的ActionSupport 基  类，实现登入的控制跳转 |
|  | RegisterAction | 封装 Spring MVC 2 的ActionSupport 基类，实现注册的控制跳转 |

## **3.3 业务逻辑层**

业务逻辑层为系统的核心层，提供了大量业务服务组件，负责处理控制层发送过来的业务数据，系统总绝大部分业务处理都在该层实现。该层实现了各种逻辑判断，即业务逻辑的封装，实现各种需求功能，并将处理后的数据传输给控制层，再由控制层进行简单处理。如果需要进行数据库操作，则调用数据持久层进行数据库相关操作。

业务逻辑层功能包含：

\* 实现各种业务处理逻辑或处理算法，比如业务信息查询、物流信息查询等等。

\* 向控制层返回处理数据信息。

\* 向持久层发送数据操作的请求，进行对数据信息的增删改查操作。

\* 作为控制层的服务层，提供接口供周边系统调用。

业务逻辑层主要包含下面的组件：

1.业务逻辑处理：负责处理各类业务逻辑的 service 组成，有关页面的跳转可以从 Spring MVC 的配置文件中看到，由于系统业务逻辑层的实现类较多，此处列举一些核心的实现类，如表 3-4 所示。

表 3-4 业务逻辑层包部分结构

|  |  |  |
| --- | --- | --- |
| 包名 | 类名 | 说明 |
| com.KunpengOrganization. service | BasicInfoService | 基本信息查询相关模块的service 接口 |
|  | UserService | 登入、注册等模块 service 接口 |
|  | RegisterService | 注册模块service接口 |
|  | LoginService | 登录模块service接口 |
| com.KunpengOrganization.serviceI  mpl | BasicInfoServiceImpl | 基本信息查询相关模块的service接口的实现类 |
|  | UserServiceImpl | 登入、注册等模块 service接口的实现类 |
|  | RegisterServiceImpl | 历史模块service接口的实现类 |
|  | LoginServiceImpl | 物流查询模块service接口的实现类 |

## **3.4 数据持久层**

在该批发社团门户网站中，数据持久层负责数据的持久操作，例如和数据库交互与数据库进行连接交互。该层抽象和封装了所有对持久化存储介质的访问。数据持久层可以访问任何的数据，除了数据库，还包括缓存数据等。在该项目框架的搭建中，是通过 Mybatis 来访问数据库。

数据持久层包含功能：

\* 对数据库中的用户信息数据进行增、删、改、查操作。

\* 对数据库中的历史信息、物流信息等信息数据自动进行增、删、改、查操作。

\* 实现事务处理，保证数据读写正常。

\* 实现意外错误操作的数据恢复。

数据持久层包结构如表 3-5 所示。

表 3-5 数据持久层部分结构表

|  |  |  |
| --- | --- | --- |
| 包名 | 文件名 | 说明 |
| com.KunpengOrganization. domain | Activity.java | 为数据库中Activity表提供 java 类进行分装 |
|  | User.java | 为数据库中User表提供 java 类进行分装 |
|  | Notice.java | 为数据库中Notice表提供 java 类进行分装 |
|  | BasicInfo.xml | 为BasicInfo类与数据库中BasicInfo表编写映射文件 |
|  | Activity.xml | 为Activity类与数据库中Activity表编写映射文件 |
|  | User.xml | 为User类与数据库中User表编写映射文件 |
|  | Notice.xml | 为Notice类与数据库中Notice表编写映射文件 |
| com.KunpengOrganization.dao | IBaseDAO.java | 数据库操作接口，所有对数据库访问和处理操作的具体实现类必须为其子类 |
|  | IBasicInfoDao.java | 基本信息数据操作接口 |
|  | INoticeDao.java | 历史数据操作接口 |
| com.KunpengOrganization.dao.impl | BaseDAOImpl.java | 实现对数据库的基本操作接口 |
|  | BasicInfoDaoImpl.java | 实现基本信息数据操作接口 |
|  | NoticeDaoImpl.java | 实现历史数据操作接口 |

## **3.5 域模型层**

在 J2EE 应用中，利用 Mybatis 在域模型基础之上来实现对数据的增、删、改、查。

域模型层包含功能：

为数据库中的表与数据实现一种数据载体，是数据库中表中数据到 Java 对象的一种映射。域模型层见表 3-6。

表 3-6 域模型层列表

|  |  |
| --- | --- |
| 名称 | 说明 |
| Activity.java | Activity 表到 Activity类的映射，用于存放社团活动信息。 |
| User.java | User 表到 User类的映射，用于存放用户的信息。 |
| Notice.java | Notice表到 Notice类的映射，用于存放历史记录。 |

## **3.6 接口设计**

### 3.6.1 内部接口

Spring 接管 Spring MVC 配置

1. 在 Spring MVC.xml，通过配置属性，将Spring MVC交给Spring Boot管理，也就是由 Spring 容器来管理 Bean，包括 Spring MVC 的 Action。

2. 在web.xml，添加配置 applicationContent.xml，需要添加一个<content-param>标签指明添加 applicationContent.xml 文件，具体见图3-4。（图中显示的是由模式匹配的方式查找符合要求的文件）。

3. Spring 接管 Mybatis 配置

向 applicationContent.xml 配置文件添加名为 sessionFactory 的 bean。

# **4. 功能模块设计**

## **4.1 功能模块设计总述**

本系统为社团门户网站，主要包括的模块为两部分：KunpengOrganization客户端子系统功能模块图、SPM 管理端子系统功能模块图，如图 4-1 所示。其中，图 4-2 是KunpengOrganization 客户端子系统功能模块图，图 4-3 是 KunpengOrganization管理端子系统功能模块图

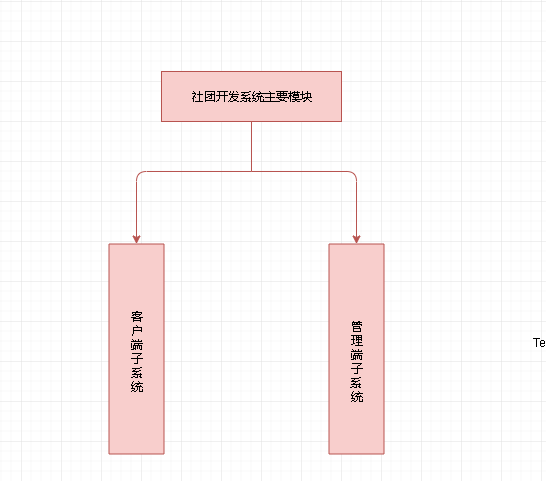


图 4-1 KunpengOrganization主要模块图

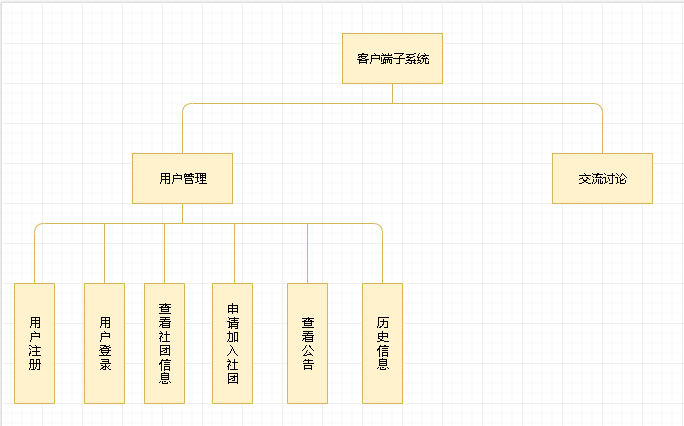


图 4-2 KunpengOrganization 客户端子系统功能模块图

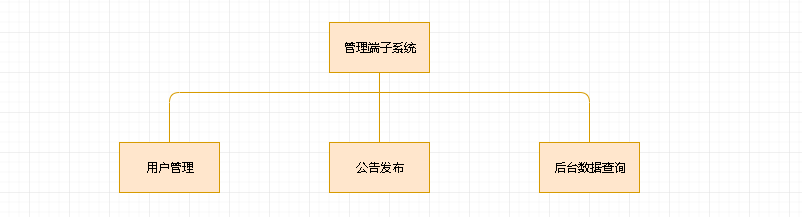


图 4-3 KunpengOrganization管理端子系统功能模块图

## **4.2 客户端子系统模块**

### 4.2.1 模块 SCM1：主页

编号：SCM1

模块名称：主页

功能简介：本模块为系统的主页模块。本模块提供新闻、通知公告等功能。并提供其他功能入口。

输入：鼠标点击事件

输出：调用的页面

### 4.2.2 模块 SCM2：社团信息

编号：SCM2

模块名称：社团信息

功能简介：本模块为社团信息模块，用户或游客可以点击进入活动后可在此浏览社团信息。

输入：无

输出：社团详细信息

### 4.2.3 模块 SCM3：登录模块

编号：SCM3

模块名称：登录模块

功能简介：本模块为系统登录模块，即用户登录系统的入口。在此模块中，用户输入自己的用户名和密码，系统在后台数据库进行查询操作后，返回布尔值，表示该输入是否正确，输入正确则进入系统，错误则对用户进行相应提示。

输入：用户名，密码

输出：用户是否登录成功

### 4.2.4 模块 SCM4：注册模块

编号：SCM4

模块名称：系统注册

功能简介：本模块为系统注册模块。用户首次进入系统时。可通过本模块进行注册。在此模块中，系统显示注册界面，用户输入相关必要的身份信息，单击确定。若注册成功，系统将以学生学号作为账号，用户设定密码作为密码存入后台数据库。注册完成后，用户可使用注册成功的账号和密码登录系统。

输入：用户名，密码

输出：用户是否注册成功

备注：对于未登录系统的游客用户，系统将自动限制一部分功能的显示

### 4.2.5 模块 SCM5：历史记录

编号：SCM5

模块名称：历史记录

功能简介：本模块为历史模块。注册的用户进入系统后。可通过本模块进行历史活动记录浏览。在此模块中，系统显示历史活动的名称、举行时间和举行人数，选择后。加入，并显示历史活动详细信息。

输入：鼠标点击事件

输出：显示历史活动

备注：对于未登录系统的游客用户，该模块不可使用

### 4.2.6 模块 SCM6：活动信息

编号：SCM6

模块名称：活动信息

功能简介：本模块为活动模块。用户或游客进入系统后，可通过本模块进行历史。在此模块中，系统显示活动信息，登录用户可以对活动进行评论。

输入：鼠标点击事件、活动评论（非必须）

输出：活动信息、活动评论

### 4.2.7 模块SCM7：公告

编号：SCM7

模块名称：公告

功能简介：本模块为公告。用户进入系统后。可查看管理员发布的公告。

输入：无

输出：公告信息

### 4.2.8 模块SCM8：社团加入

编号：SCM8

模块名称：加入社团

功能简介：本模块为加入社团。登录用户进入系统后，可填写自己的信息，加入社团。如果加入失败，系统将出现提示。

输入：鼠标点击事件、用户信息

输出：是否成功

## **4.3 管理端子系统模块**

### 4.3.1 模块DAM1：用户管理

编号：DAM1

模块名称：用户管理

功能简介：管理员进入“用户管理”模块

管理员选择查看系统中已经存在的用户信息。系统显示全部用户信息，并提供删除，修改，查询操作。用户选择相应的用户进行删除，修改，并确认操作。系统更新用户列表。用户输入部分关键字，选择查询操作。系统显示所有符合的用户，并提供删除，修改操作。用户选择处理用户注册申请确认的功能。系统显示所有需要进行确认的用户的注册申请，并提供确认和删除操作。用户选择确认操作。

输入：鼠标点击事件

输出：无

### 4.3.2 模块DAM2：公告发布

编号：DAM2

模块名称：公告发布

功能简介：管理员发布公告

管理员在该模块输入公告，并发布。

输入：公告内容

输出：发布是否成功

### 4.3.3 模块DAM3：后台数据查询

编号：DAM3

模块名称：后台数据查询

功能简介：管理员查询后台信息

管理员在该模块管理员查询后台信息。

输入：查询的信息关键字

输出：查询的信息内容

# **5．数据库设计**

## **5.1 数据库种类及特点**

本项目采用了MySQL 8.0.17的数据库。MySQL是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 公司，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL具有以下几个特性：

（1） MySQL为多种编程语言提供了 API。这些编程语言包括 C、C++、Python、

Java、Perl、PHP、Eiffel、Ruby,.NET和 Tcl 等。

（2） 优化的 SQL查询算法，有效地提高查询速度。

（3） 提供多语言支持，常见的编码如中文的 GB 2312、BIG5，日文的 Shift\_JIS

等都可以用作数据表名和数据列名。

（4） 提供 TCP/IP、ODBC 和 JDBC等多种数据库连接途径。

（5） 提供用于管理、检查、优化数据库操作的管理工具。

（6） 支持大型的数据库。可以处理拥有上千万条记录的大型数据库。

（7） MySQL 使用标准的 SQL数据语言形式。

## **5.2 数据库逻辑结构**

鲲鹏素拓社团门户的实体关系图（概念数据模型）如下图所示。

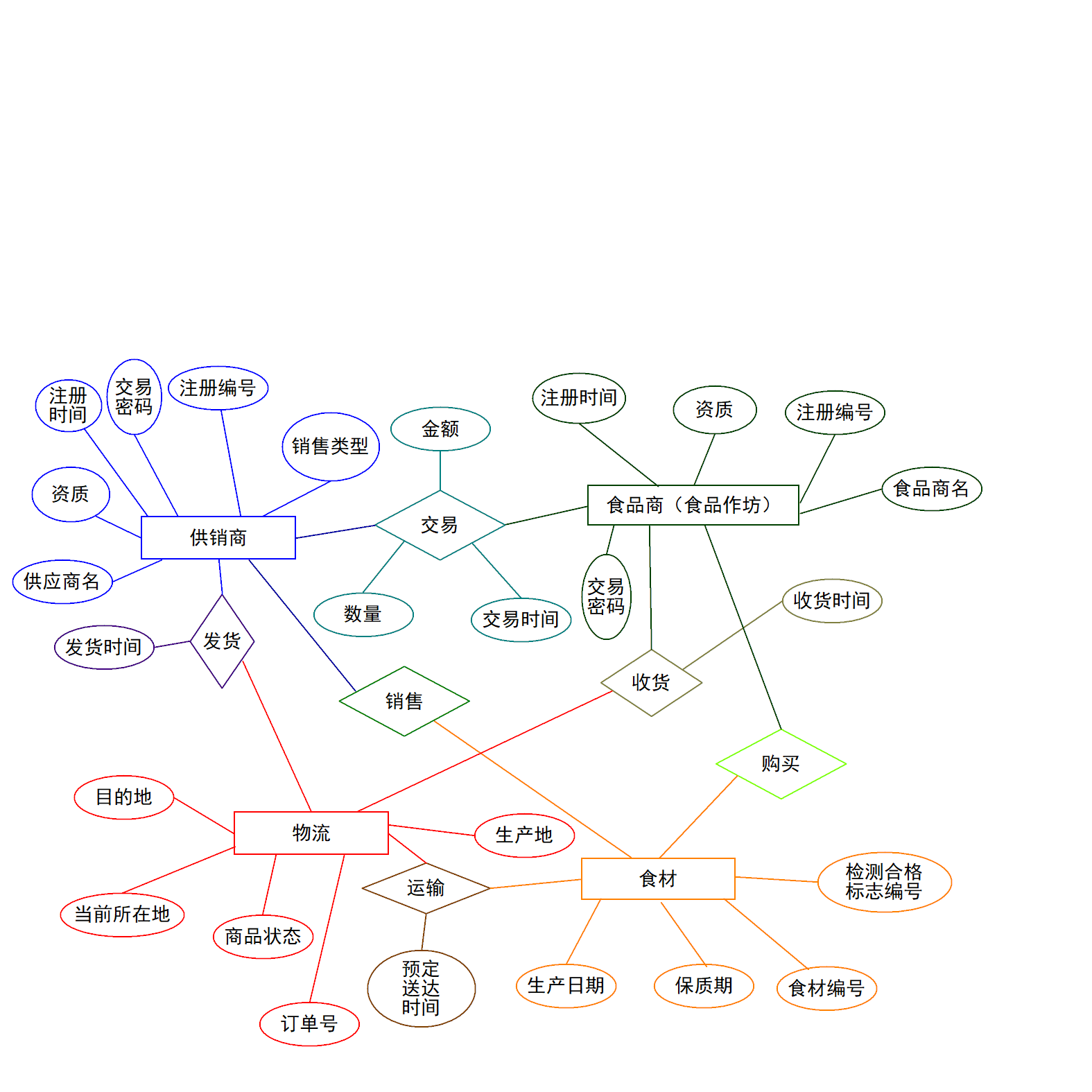


图 5-1 实体关系图

## **5.3 物理结构设计**

根据上面的实体关系，设计数据库表即根据物理数据模型图5-1可知，系统一共应有5张表。

### 5.3.1 Customer表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Customer表 |  |  |
| 字段名 | 字段代码 | 字段类型 | 可否为空 | 备注 |
| 用户编号 | customer\_id | int (11) | no | 主键，自动生成 |
| 用户名 | user\_real\_name | varchar(40) | no |  |
| 注册时间 | register\_date | varchar(30) | no | 自动生成 |
| 登录密码 | login\_Password | varchar(100) | no |  |
| 用户身份 | status | int(2) | no | 只能取值0、1 |

### 5.3.2 Auditor表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Auditor表 |  |  |
| 字段名 | 字段代码 | 字段类型 | 可否为空 | 备注 |
| 发行人员编号 | auditor\_id | int (11) | no | 主键，自动生成 |
| 发行人员姓名 | auditor\_real\_name | varchar(40) | no |  |
| 注册时间 | register\_date | varchar(30) | no | 自动生成 |
| 登录密码 | auditor\_password | varchar(100) | no |  |

### 5.3.3 Buyer表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Buyer表 |  |  |
| 字段名 | 字段代码 | 字段类型 | 可否为空 | 备注 |
| 采购人员编号 | buyer\_id | int (11) | no | 主键，自动生成 |
| 采购人员姓名 | buyer\_real\_name | varchar(40) | no |  |
| 注册时间 | register\_date | varchar(30) | no | 自动生成 |
| 登录密码 | buyer\_password | varchar(100) | no |  |

### 5.3.4 Textbook表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Textbook表 |  |  |
| 字段名 | 字段代码 | 字段类型 | 可否为空 | 备注 |
| 书号(ISBN) | isbn | varchar(20) | no | 主键 |
| 书名 | book\_name | varchar(100) | no |  |
| 出版社 | publisher | varchar(100) | no |  |
| 价格 | price | double | no | 默认值为0 |

### 5.3.5 Stock表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Stock表 |  |  |
| 字段名 | 字段代码 | 字段类型 | 可否为空 | 备注 |
| 库存编号 | stock\_id | int (11) | no | 主键，自动生成 |
| 书号(ISBN) | book\_isbn | varchar(20) | no | 外键 |
| 书名 | book\_name | varchar(100) | no |  |
| 库存量 | store\_count | int(11) | no | 默认值为0 |

### 5.3.6 Out\_of\_Stock表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Out\_of\_Stock | 表 |  |  |
| 字段名 | 字段代码 | 字段类型 | 可否为空 | 备注 |
| 出库编号 | out\_id | int (11) | no | 主键，自动生成 |
| 书号(ISBN) | book\_isbn | varchar(20) | no | 外键 |
| 出库量 | out\_count | int(11) | no |  |
| 出库时间 | date\_of\_out | varchar(30) | no |  |

### 5.3.7 In\_Storage表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | In\_Storage表 |  |  |
| 字段名 | 字段代码 | 字段类型 | 可否为空 | 备注 |
| 入库编号 | storage\_id | int (11) | no | 主键，自动生成 |
| 书号(ISBN) | book\_isbn | varchar(20) | no | 外键 |
| 入库时间 | date\_of\_instorage | varchar(30) | no |  |
| 入库数量 | instorage\_count | int (11) | no |  |

### 5.3.8 Textbook\_Lack表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Textbook\_Lack | 表 |  |  |
| 字段名 | 字段代码 | 字段类型 | 可否为空 | 备注 |
| 编号 | lack\_id | int (11) | no | 主键，自动生成 |
| 缺书书号 | book\_isbn | varchar(20) | no | 外键 |
| 缺书本数 | required\_count | int (11) | no |  |
| 是否已处理 | lack\_status | int(2) | no | 只能取值0、1 |

### 5.3.9 Textbook\_Purchase表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Textbook\_Purchase | 表 |  |  |
| 字段名 | 字段代码 | 字段类型 | 可否为空 | 备注 |
| 主键字段 | primary\_column | int (11) | no | 主键 |
| 消费者编号 | customer\_id | int (11) | no | 外键 |
| 发行人员编号 | auditor\_id | int (11) | no | 外键 |
| 购书书号 | book\_isbn | varchar(20) | no | 外键 |
| 购书本数 | purchase\_count | int (11) | no |  |
| 订单号 | order\_id | char(40) | no |  |
| 购买日期 | purchase\_date | varchar(30) | no |  |

### 5.3.10 Textbook\_Order表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Textbook\_Order | 表 |  |  |
| 字段名 | 字段代码 | 字段类型 | 可否为空 | 备注 |
| 主键字段 | primary\_column | int (11) | no | 主键 |
| 购书书号 | book\_isbn | varchar(20) | no | 外键 |
| 消费者编号 | customer\_id | int (11) | no | 外键 |
| 购书本数 | purchase\_count | int (11) | no |  |
| 订单号 | order\_id | char(40) | no |  |
| 是否已处理 | order\_status | int(2) | no | 只能取值0、1 |
| 提交日期 | submit\_date | varchar(30) | no | 自动生成 |
| 处理日期 | deal\_date | varchar(30) | yes |  |

# **6. 界面设计**

## **6.1 首页设计**

首页设计如下图所示



图 6-1 界面参考

## **6.2 登录注册**

登录注册设计如下图所示

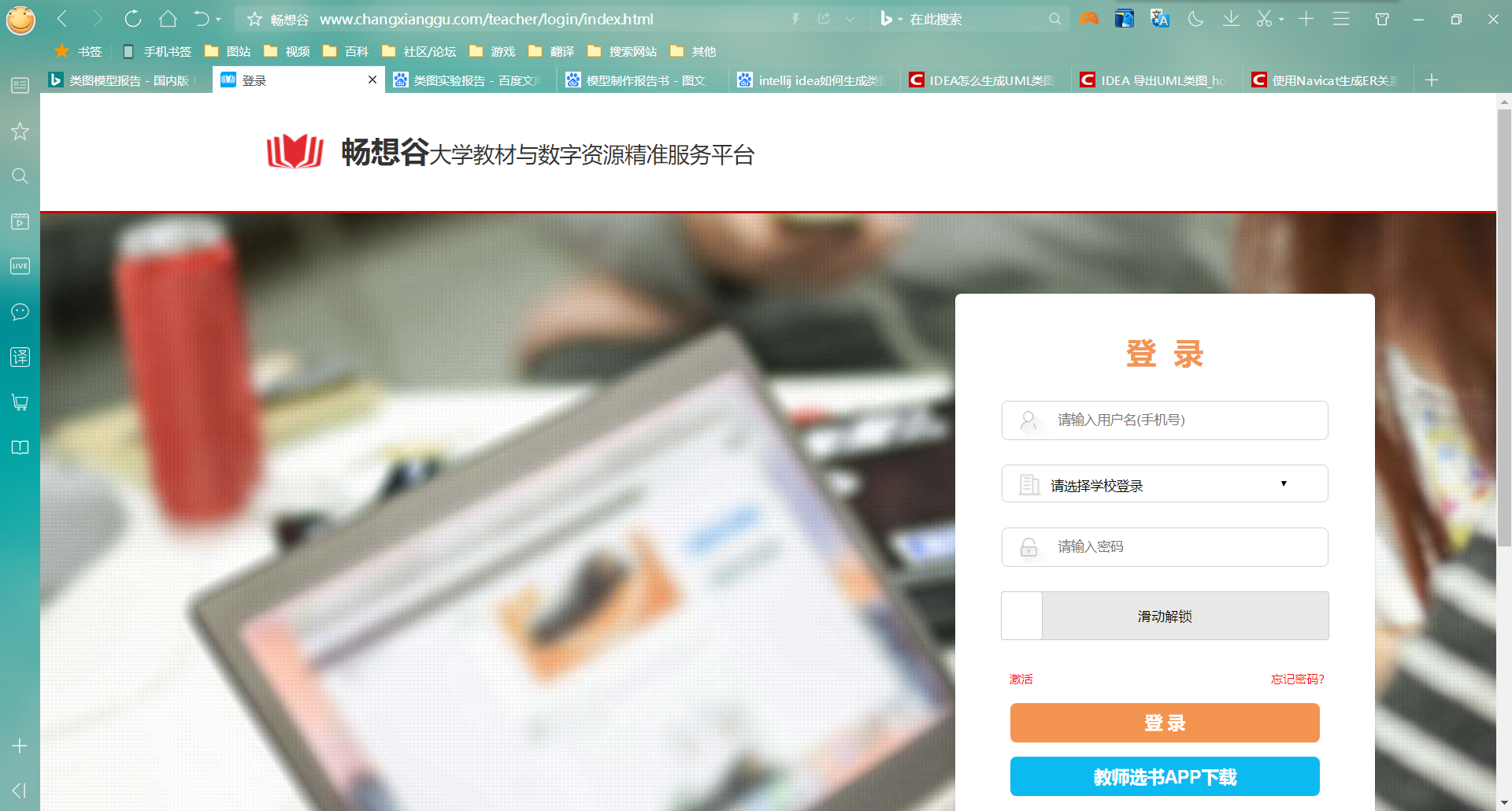


图 2-2 登录注册参考