**《论坛管理系统项目设计书》**

**—第一阶段性报告**

**课程名称**：《软件课程设计Ⅰ》

**项目名称**：《论坛管理系统》

**指导老师**：周 涛

**项目组编号**：40

**学生姓名**：沈卓诺 **学 号**：922106840431

**学生姓名**：杨靖翔 **学 号**：922106840441

**学 院**：计算机科学与工程学院 **专 业**：计算机科学与技术

**班 级**：9221062302班

**提交日期：**2024年7月20日

1. 系统需求

20至21世纪是计算机行业逐渐成熟、互联网和信息技术（IT）行业蓬勃发展的重要时期。在这一阶段，随着信息技术的迅猛发展，各类交流平台相继涌现，极大地促进了人们之间的沟通与资源共享。知乎、百度贴吧、博客园、CSDN等论坛交流平台的兴起，不仅为用户提供了丰富的信息和知识交流渠道，也推动了社会各界对技术和思想的深入探讨。这些平台的多样性和互动性，使得信息传播更加高效，为个人与团体的成长提供了新的机遇。因此，本项目组希望设计更简易的论坛平台模板，使得其他开发者更轻松地开发，无计算机基础的人们也能轻易上手管理设计，从而进一步推进人们之间的沟通与资源共享，发扬互联网精神。

1. 功能需求
   1. 系统角色说明

通过对使用网页端论坛如知乎、掘金和CSDN的主要用户群体进行深入分析，可以有效地对用户进行分类。根据用户的年龄差异，可以将其划分为青少年、青年、中年和老年四个类别；而依据用户对资讯内容的偏好，则可进一步细分为新闻、科技、体育、娱乐等多个领域。此外，考虑到用户在使用论坛时的不同期望，可以将其系统角色定义为四种类型，具体用户与角色的对应关系详见图 1所示。

1. 博主

这类用户是平台上贡献内容的创作者，他们可能是专业作者、自媒体从业者或有一定影响力的网红。博主可以进一步细分为专栏作者、视频创作者、自媒体博主等，针对其具体的创作形式和需求提供相应的支持和服务。

1. 游客

这类用户是平台上的访问者，主要是浏览资讯内容的用户，可能不具备内容创作的能力，但对平台上的内容感兴趣。游客可以进一步细分为注册用户和未注册用户，根据其行为习惯和需求针对性地提供服务和推荐内容。

1. 广告商

这类用户是希望在资讯平台上投放广告的商家或广告代理公司，他们通过在平台上购买广告位来推广产品或服务。广告商可以进一步细分为线上广告商、线下广告商、品牌广告商等，根据其需求和预算量身定制广告解决方案。

1. 管理人员

这类用户是资讯平台的运营和管理人员，负责平台内容审核、用户管理、技术支持等工作。管理员需要具备良好的专业知识和沟通能力，保障平台的正常运营和用户体验。

图 1

1.2系统角色具体说明

系统各个角色对面向网页端的论坛的整体需求如表 1所示

|  |  |  |
| --- | --- | --- |
| 用户群 | 角色 | 需求 |
| 游客，普通用户 | 游客 | 寻找信息：这些游客可能在论坛中寻找特定主题或关键词的信息，比如旅游目的地的介绍、景点推荐、交通指南等。 |
| 参与讨论：虽然不发帖，但这些游客可能会浏览其他用户发表的帖子，阅读评论和讨论，以获取更多观点和见解。 |
| 寻求建议：他们可能会寻求其他论坛用户的建议和意见，比如寻找旅行建议、住宿推荐等。 |
| 解决问题：如果他们遇到困惑或问题，可能会在论坛中寻找相关信息或解决方案。 |
| 发帖用户，博主 | 博主 | 内容展示：博主希望能够在论坛中展示自己的专业知识或兴趣内容，吸引更多读者关注和阅读自己的帖子或文章。 |
| 互动交流：博主渴望与读者互动，回答他们的问题，交流意见，获得反馈和建议，建立良好的读者关系。 |
| 提升影响力：通过在论坛中发布高质量内容和参与讨论，博主希望提升自己在相关领域的影响力和专家形象，吸引更多粉丝和读者。 |
| 获取支持和合作机会：博主可能在论坛中寻求与其他博主或相关品牌合作的机会，推广自己的内容或产品，从而扩大影响力和盈利渠道 |
| 线上广告商、线下广告商、品牌广告商 | 广告商 | 宣传和推广：广告商希望通过论坛在目标受众面前进行品牌和产品的宣传推广，吸引潜在客户的注意。 |
| 提高知名度：广告商希望借助论坛的广告平台提升自身品牌知名度和曝光度，从而扩大市场份额和影响力 |
| 目标定位：广告商希望能够在论坛上精准地定位到他们的目标受众群体，以实现有效的推广效果和投资回报率。 |
| 管理人员 | 管理人员 | 管理系统用户信息，审核注册认证信息 |

表 1

2需求规定

根据对主流论坛网站及其市场的分析，论坛作为一个在线交流平台，论坛需要具有用户注册和登录功能，用户可以注册账号，填写个人信息，并通过用户名和密码登录账号。注册后用户可以发布新帖子，并在其他帖子下回复评论。帖子可以按主题分类，以便用户更容易找到感兴趣的内容。同时要具有搜索功能，使用户可以通过关键词搜索帖子或用户，以查找相关信息。注册用户需要能够管理个人资料，如可以编辑个人资料，包括头像、个人简介等信息。除此之外，还需要具备通知功能，用户可以收到与自己相关的消息通知，比如有人回复了自己的帖子或评论。

2.1需求规定具体分析

（1）需求规定一：账户管理

【1】功能说明

用户需要系统提供账户管理功能，主要包括账户注册、登录、登出、账户信管理等通用功能。

【2】具体功能实现

用户注册：博主和游客在使用系统前首先要为自己注册一个账号，设置用户名与密码等基本注册信息并提交注册申请后等待服务器返回验证码即可注册。管理员与广告商由后台设置则无需注册。

用户登录：用户使用系统时，需要输入用户名和密码进行登录操作。用户登出：用户退出当前账号，退出系统。

账号管理：用户可以自行修改昵称，头像以及密码。

【3】功能点分析

校验功能点：使用正则表达式在前端校验账号，昵称，密码是否符合规范，如果不符合规范则不向后端发送请求。在请求邮箱验证码前需要通过图片验证码，防止恶意注册。

邮箱验证功能点：注册账号需要向服务器发送请求，由服务器发送邮件用来验证邮箱真实性

信息保存功能点：勾选记住我选项时为用户在浏览器保存加密的md5密码Cookie，下次登录时自动输入密码与账号

【4】交互形式

点击网站登录/注册按钮弹出登录/注册小窗口，可以在窗口点击连接跳转至其他登录/注册/重置密码小窗口，窗口会保存部分表单信息并刷新图片验证码。

（2）需求规定二：修改/发布帖子

【1】功能说明

“发帖”是一种在网络论坛、社交媒体、博客或其他在线平台上分享信息、观点或内容的功能。用户可以撰写文本、上传图片、视频或其他媒体。发帖可以选择板块，方便其他用户快速找到相关内容。可视化编辑：许多平台提供可视化编辑器，用户可以方便地排版、加粗、插入链接等，提升内容的可读性。

【2】具体功能实现

可视化编辑器：编辑器为用户提供了排版、加粗、插入链接等内容，用户可以在设置栏内为帖子设置封面，标题，添加附件及下载所需积分，设置板块分类并发布。

帖子编辑：用户可以对已发布帖子进行修改，更改文章内容。

【3】功能点分析

编辑器功能点：网站为用户提供了markdown编辑器以及富文本编辑器，同时支持对两种编辑器的切换。支持以一种编辑器打开另一种类型的文章。

图片功能点：支持在文本内插入图片，支持插入封面图片及其缩略图，支持在编辑时全屏预览图片

【4】交互形式

登录后点击发帖按钮即可进入文本编辑模式，纂写完内容后即可发表。

（3）需求规定三：评论、点赞与排序

【1】功能说明

用户可以对他人的帖子或评论进行点赞，表示对内容的认可和喜爱。可以对帖子发表看法、提问或与其他用户进行交流，形成二次讨论。帖子和评论可以按照不同标准进行排序，比如时间、点赞数、回复数等。

【2】具体功能实现

评论点赞：支持一级评论与二级评论，用户登录后即可在帖子下方进行评论，被评论/点赞者会收到被评论的消息提醒

排序：支持按时间排序与按点赞排序，不同的排序下帖子的呈现方式不同

【3】功能点分析

置顶功能点：管理员在板块中可以置顶多个文章，博主可以在自己的帖子中置顶一个评论。

【4】交互形式

用户登录后即可与帖子交互，未登录情况下也可以使用排序功能并浏览帖子

2.2业务实体分析

（1）主要实体

|  |  |
| --- | --- |
| 序号 | 实体 |
| a | 用户（User） |
| b | 帖子（Article） |
| c | 评论（Comment） |
| d | 多媒体文件（MediaFile） |
| e | 日志记录（LogEntry） |

（2）实体属性

1. 用户（User）

（1）用户名

（2）密码

（3）电子邮箱：用于验证用户身份、接收通知或重置密码。

（4）注册日期：用户在论坛注册的时间，通常用于区分新老用户。

（5）头像：用户在论坛中的个人头像，通常可以自定义。

2. 帖子

（1）帖子ID

（2）标题

（3）内容

（4）创建时间

（5）作者ID

（6）回复数

（7）点赞数

（8）浏览量

3. 评论（Comment）

（1）评论ID

（2）内容

（3）创建时间

（4）评论者ID

（5）点赞数

（6）帖子ID

（7）父评论ID

4. 多媒体文件

（1）文件ID

（2）文件名称

（3）文件类型

（4）上传日期

（5）所属用户ID

5.日志记录

（1）日志ID

（2）用户ID

（3）操作类型

（4）操作详情

（5）操作日期

3非功能需求

1 性能需求

响应时间：页面加载时间应小于3秒。

并发用户支持：系统应支持至少1000个并发用户。

2可用性需求

用户友好的界面：设计应简洁直观，方便用户操作。

可访问性：确保所有用户都能访问。

数据完整性：确保在系统故障或崩溃时不丢失关键数据。

3安全性需求

用户数据保护：使用md5加密存储用户密码和敏感信息。

4可维护性：

代码可读性和模块化：代码应保持良好的注释和结构，以便后续维护。

系统监控日志：记录关键操作和错误信息，便于排查和修复问题。

5扩展性：

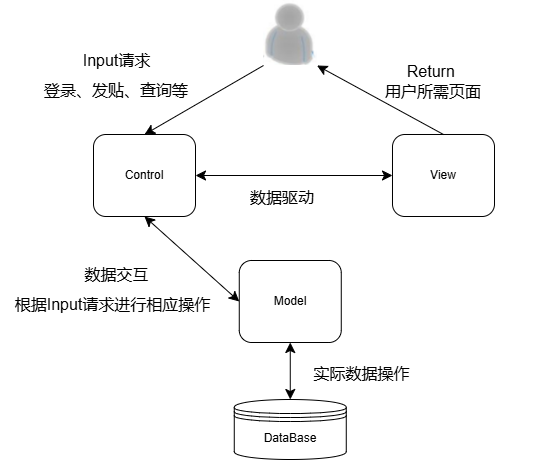
模块化设计：系统架构应支持后续功能的扩展和模块的增加。

外部接口扩展：能够轻松地扩展数据库设计以支持新的功能或数据类型。

1. 系统设计
2. **基于MVC的软件架构设计**
   1. **架构设计层次分析**

论坛管理系统的设计工作是整个开发阶段中极为重要的一环。在这一阶段，系统架构的设计显得尤为关键，因为它直接关系到系统的整体性能与可维护性。系统架构设计涵盖了多个方面，主要包括用户操作界面的设计、业务逻辑处理模块的构建以及数据信息的存储策略等。

为了提升系统的可扩展性和灵活性，本项目拟采用MVC（三层架构）设计模式来实现软件架构。MVC模式通过将系统分为模型（Model）、视图（View）和控制器（Controller）三个核心部分，有效地将业务逻辑、用户界面和数据处理分开，以优化代码组织结构，支持项目组成员之间进行并行开发，并减少因代码耦合而产生维护困难。



MVC设计模式结合三层结构，可以对系统进行进一步组织划分，以优化代码的组织结构。具体来说，Controller和View因代码耦合度高，可以合并为表现层（视图层view），Model可根据调用方式与具体使用功能划分为业务逻辑层（服务层service）和数据访问层（持久层Dao）。各层的功能与具体内容如下：

1. 表现层：

表现层负责用户与软件之间的交互，旨在为用户提供一种更直观、易于接受和使用的交互界面，以实现更加流畅和灵活的用户体验。

具体来说，表现层的view部分负责系统数据的展示，并提供相应的接口供用户输入数据，同时对原始数据进行基本的验证和封装。表现层的Controller部分则负责接收用户的输入请求，并对接收到的数据进行格式校验。对于格式正确的数据，Controller会通过调用业务逻辑层的API执行页面控制、数据运算、决策选择及异常处理等业务规则和业务逻辑，从而实现与Model的交互，更新数据库和处理数据。

因此，表现层与业务逻辑层之间存在依赖关系，必须规范统一的数据传输格式和数据文件格式。我们计划以JSON格式为主，辅以XML格式进行数据信息传输。

1. 业务逻辑层：

业务逻辑层实现业务的主要逻辑，是系统架构中体现核心的部分。

实现业务的主要逻辑，是系统架构中体现核心价值的部分。将一个业务中所有的操作封装成一个方法，同时保证方法中所有的数据库更新操作，即保证同时成功或同时失败。避免部分成功部分失败引起的数据混乱操作。

业务逻辑层是软件架构中的核心部分，该层位于界面控制层和数据访问层之间，主要负责管理业务流程的启动、执行和终止等阶段，对用户的输入数据进行验证以确保数据的正确性和完整性，并在数据从交互界面传递到数据访问层之前进行必要的数据转换和数据格式化，同时为系统提供业务逻辑的接口，实现系统的业务逻辑，对事务进行控制，以及对外提供或调用 Web 服务。该层的逻辑代码具有高内聚性，每个模块或者服务都只关注一个特定的业务领域。该逻辑层主要与技术服务层和数据层存在依赖关系，层次间的通信主要采用 Web 服务和 API 调用的方式。

1. 数据访问层：

也称为是持久层，其功能主要是负责数据库的访问（可以访问数据库、二进制文件、文本文件等），是对数据库，而不是对数据的操作。简单的说法就是实现对数据表的Select，Insert，Update，Delete的操作。如果要加入ORM的元素，那么就会包括对象和数据表之间的mapping，以及对象实体的持久化。也就是哪个类对应哪个表，哪个属性对应哪个列。持久层的目的就是，完成对象数据和关系数据的转换。

**（2）业务逻辑层**

业务逻辑层是软件架构中的核心部分，该层位于界面控制层和数据访问层之间，主要负责管理业务流程的启动、执行和终止等阶段，对用户的输入数据进行验证以确保数据的正确性和完整性，并在数据从交互界面传递到数据访问层之前进行必要的数据转换和数据格式化，同时为系统提供业务逻辑的接口，实现系统的业务逻辑，对事务进行控制，以及对外提供或调用 Web 服务。该层的逻辑代码具有高内聚性，每个模块或者服务都只关注一个特定的业务领域。该逻辑层主要与技术服务层和数据层存在依赖关系，层次间的通信主要采用 Web 服务和 API 调用的方式。

**（3）技术服务层**

技术服务层是软件架构中的关键抽象层，它位于业务逻辑层和数据访问层之间，通过将业务逻辑和底层技术分离，从而降低系统各部分之间的耦合度，从而提高系统的可维护性、可拓展性和可测试性。同时，技术服务层的抽象性表现在其将底层技术细节抽象化，为业务逻辑层提供简化且统一的接口，并能够封装缓存、消息队列和数据库访问等底层服务和中间件调用。该层包含由系统开发平台和第三方产品提供的各类基础模块，并与数据层存在依赖关系，层次间的通信主要采用API调用的方式。

**（4）数据层**

数据层也称为模型层，主要提供存储数据实体，向业务逻辑层提供访问数据库、文件系统等资源的接口，对于整个应用程序的性能、可维护性和可扩展性至关重要。该层封装了所有数据实体和数据库的交互逻辑，隐藏了数据存储的细节，并负责数据的存储和检索，包括数据的创建、读取、更新和删除等操作，同时还提供了数据同步、数据备份和数据恢复等服务。该逻辑层主要与基础架构服务层存在依赖关系，层次间的通信主要采用Web服务和API调用的方式。

**（5）基础架构服务层**

基础架构服务层主要包含应用服务器、数据库等，是一组为MVC架构中的视图、控制器和模型提供支持的底层服务和组件，为上层组件提供了简化的接口等基础设施。该服务层确保了应用程序的稳定性，使得开发者可以专注于业务逻辑的实现，而不必过多地关注底层技术细节，为应用程序的长期发展和演化打下了坚实的基础。

* 1. **系统软件架构设计**

1. **功能模块设计**

**2.1模块功能划分**

**2.2模块设计**

**（1）功能点一：账户管理**

**【1】基于MVC模式的资源表示**

**1. Controller 表**

| **序号** | **控制器** | **描述** |
| --- | --- | --- |
| **1** | **UserController** | **处理用户注册、登录、登出及管理逻辑** |
| **2** | **ProfileController** | **处理用户个人资料的查看和更新** |
| **3** | **PasswordController** | **处理用户密码重置和修改逻辑** |
| **4** | **AdminController** | **管理员账号操作及管理逻辑** |

**2. View 表**

| **序号** | **视图** | **描述** |
| --- | --- | --- |
| **1** | **register.html** | **用户注册页面** |
| **2** | **login.html** | **用户登录页面** |
| **3** | **profile.html** | **用户个人资料页面** |
| **4** | **changePassword.html** | **修改密码页面** |
| **5** | **userList.html** | **用户列表（管理员视图）** |
| **6** | **resetPassword.html** | **重置密码页面** |

**3. Model 表**

| **序号** | **模型** | **描述** |
| --- | --- | --- |
| **1** | **User** | **表示用户基本信息的模型** |
| **2** | **UserProfile** | **处理用户个人资料的模型** |
| **3** | **UserCredentials** | **处理用户认证信息（密码）的模型** |
| **4** | **Admin** | **表示管理员帐户信息的模型** |

**4. DAO 表**

| **序号** | **DAO** | **描述** |
| --- | --- | --- |
| **1** | **UserDAO** | **处理用户数据访问的对象** |
| **2** | **ProfileDAO** | **处理用户资料数据访问的对象** |
| **3** | **CredentialsDAO** | **处理用户认证信息数据访问的对象** |
| **4** | **AdminDAO** | **处理管理员数据访问的对象** |

**【2】基于UML序列图的业务逻辑**

**（2）功能点二：修改/发布帖子**

**【1】基于MVC模式的资源表示**

**1. Controller 表**

| **序号** | **控制器** | **描述** |
| --- | --- | --- |
| **1** | **PostController** | **处理帖子发布、修改、删除和获取逻辑** |
| **2** | **CommentController** | **处理评论的添加、删除和获取逻辑** |
| **3** | **CategoryController** | **处理帖子分类的管理逻辑** |
| **4** | **TagController** | **处理帖子标签的管理和分配逻辑** |

**2. View 表**

| **序号** | **视图** | **描述** |
| --- | --- | --- |
| **1** | **createPost.html** | **帖子发布页面** |
| **2** | **editPost.html** | **帖子修改页面** |
| **3** | **postDetail.html** | **显示单个帖子的详细内容页面** |
| **4** | **postList.html** | **显示所有帖子的列表页面** |
| **5** | **commentSection.html** | **显示评论及添加评论的界面** |

**3. Model 表**

| **序号** | **模型** | **描述** |
| --- | --- | --- |
| **1** | **Post** | **表示论坛帖子的内容和基本信息** |
| **2** | **Comment** | **表示评论的内容和相关信息** |
| **3** | **Category** | **表示帖子分类的信息** |
| **4** | **Tag** | **表示与帖子关联的标签信息** |

**4. DAO 表**

| **序号** | **DAO** | **描述** |
| --- | --- | --- |
| **1** | **PostDAO** | **处理帖子数据访问的对象** |
| **2** | **CommentDAO** | **处理评论数据访问的对象** |
| **3** | **CategoryDAO** | **处理分类数据访问的对象** |
| **4** | **TagDAO** | **处理标签数据访问的对象** |

**【2】基于UML序列图的业务逻辑**

**（2）功能点二：点赞/评论**

**【1】基于MVC模式的资源表示**

**1. Controller 表**

| **序号** | **控制器** | **描述** |
| --- | --- | --- |
| **1** | **CommentController** | **处理评论的添加、修改、删除和获取逻辑** |
| **2** | **LikeController** | **处理点赞/取消点赞的逻辑** |
| **3** | **SortController** | **处理帖子评论的排序和筛选逻辑** |

**2. View 表**

| **序号** | **视图** | **描述** |
| --- | --- | --- |
| **1** | **commentList.html** | **显示评论列表的页面** |
| **2** | **addComment.html** | **添加评论的界面** |
| **3** | **editComment.html** | **修改评论的界面** |
| **4** | **likeButton.html** | **点赞按钮视图** |
| **5** | **commentSortOptions.html** | **评论排序和筛选选项的界面** |

**3. Model 表**

| **序号** | **模型** | **描述** |
| --- | --- | --- |
| **1** | **Comment** | **表示评论的内容、作者和时间等信息** |
| **2** | **Like** | **表示点赞的记录，包括点赞用户和对应评论** |
| **3** | **SortCriteria** | **表示评论排序的标准和选项** |

**4. DAO 表**

| **序号** | **DAO** | **描述** |
| --- | --- | --- |
| **1** | **CommentDAO** | **处理评论数据访问的对象** |
| **2** | **LikeDAO** | **处理点赞数据访问的对象** |
| **3** | **SortDAO** | **处理评论排序和筛选的数据访问对象** |

**【2】基于UML序列图的业务逻辑**

1. **数据库设计**

本小节介绍论坛管理系统的数据库设计，以确保使用MySQL数据库准确地管理论坛文件数据、用户信息数据

* 1. **逻辑结构设计**
  2. **物理结构设计**