摘 要

随着科技的快速发展和现代化的脚步不断向前，现代人的生活节奏也不断加快。在线图片、视频悬赏网站必定会受到时长的欢迎。

本文设计的千里眼悬赏论坛包括的图片、视频悬赏网站所有业务。该论坛后端使用java语言编写，前端使用html、css、javascript，模板引擎使用thymeleaf，数据库使用MySQL数据库管理系统和Redis数据库。本论坛实现了对图片、视频悬赏业务的管理，总共分为8个模块，分别为登录及注册模块、个人信息维护模块、悬赏模块、答案模块、刷一刷模块、等级模块、管理员审核模块、通知模块。本文主要描述悬赏模块、答案模块、刷一刷模块、管理员审核模块等级模块和通知模块。登录及注册模块、个人信息维护模块主要操作是对后端数据库的增删改查操作。整个系统后端框架使用Springboot、Mybatis和shiro，结构清晰、系统安全性高、维护方便。

本文的系统测试主要采用的是黑盒测试，黑盒测试目的在于检测系统各部分功能是否达到预期要求。

关键词：Java；MySql数据库；Redis数据库

Abstract

With the rapid development of science and technology and the pace of modernization, the pace of life of modern people is also accelerating. Online images and video reward sites are sure to be popular with the duration.

The Clairvoyant Reward Forum designed in this article includes all the pictures and videos on the website. The forum backend is written in Java, the front end uses html, css, javascript, the template engine uses thymeleaf, the database uses MySQL database management system and Redis database. This forum realizes the management of picture and video reward business. It is divided into 8 modules, namely login and registration module, personal information maintenance module, reward module, answer module, brush brush module, level module, administrator audit module. , notification module. This article mainly describes the reward module, the answer module, the brush-brush module, the administrator audit module level module and the notification module. The main operation of the login and registration module and the personal information maintenance module is to add, delete and change the back-end database. The entire system back-end framework uses Springboot, Mybatis and shiro, with clear structure, high system security and easy maintenance.

The system test in this paper mainly uses black box test. The purpose of black box test is to detect whether the functions of various parts of the system meet the expected requirements.

目 录

# 1 绪 论

## 选题背景

随着科技的快速发展和现代化的脚步不断向前，现代人的生活节奏也不断加快。很多人想去旅游或是回到曾经充满回忆的地方，但是因为种种原因不能如愿。在这种背景下，市面上需要一个针对于这种情况的在线图片视频悬赏网站。本论坛具有极强的针对想和强大的业务处理，灵活方便，具有极高的安全性。本论坛使用浏览器与服务器议题的系统结构的面向对象编程，严格按照MVC封层设计的思想，数据库严格按照第三范式设计，极易于维护和添加新功能。旅游是现代人极其重要的精神需求，在个人的知识储备和眼界的提升方面起到了重要的作用。在线图片视频悬赏是当前社会形势下的必然需求。因此，本论坛应运而生。

本文设计的千里眼论坛是基于Java语言所设计的B/S结构的在线图片视频悬赏网站。本论坛使用Mysql，Redis作为数据库，前端使用JQuery框架和Bootstrap组件库，后端使用Springboot、Mybatis、Shiro框架实现，使用thymeleaf作为后端解释器。本网站使用Visual Code作为前端开发工具，使用Intellij idea作为后端开发工具。本系统实现了一个及运行性能、可靠性、趣味性、安全性的一个在线悬赏论坛。

本文主要阐述千里眼论坛的展示界面、登录、注册、悬赏操作、后台管理等方面的具体实现以及相关技术。千里眼论坛包括个人信息维护、个人悬赏维护、个人答案维护、最佳悬赏答案、随机刷优秀的图片视频悬赏答案、后台管理员答案审核、在线时长统计等模块

## 选题意义

当今社会随着科技的不断进步，尤其是年轻人的工作强度也不断加重，人们属于自己的时间也不断的减少。同时目前科技信息化发展迅速，在这种背景下在线图片、视频悬赏网站能够体现出来的价值就很大。一方面能够极大的减少工作繁忙的上班族的外出时间，另一方面这种在线悬赏帮助论坛也符合当下人们的生活工作状态。同时目前市面上比较知名的面向图片、视频的悬赏网站可以说是少之又少，在这少之又少的份额里的网站开发参差不齐。这也为了本文设计的这样一款结构清晰、安全性高的在线图片、视频悬赏网站提供了生存机会的市场前景。

2.1java

## 2.2Springboot

Spring团队决定他们希望为开发人员提供一些相对自动化配置过程的实用程序，并加快构建和部署Spring应用程序的过程，因此他们发明了Spring Boot。Spring Boot是一个实用程序项目，旨在轻松构建基于Spring，生产就绪的应用程序和服务，而不用担心。它提供了使用最小的代码/配置开箱即用的Spring Web应用程序启动和运行的最短路径。有许多特定于Spring Boot的功能，如依赖管理，自动配置和嵌入式servlet容器。

为了加速依赖关系管理过程，Spring Boot隐式地为每种类型的Spring应用程序打包所需的兼容第三方依赖项，并使用starters将它们公开给开发人员。

Spingboot框架内部实现了自己的依赖关系描述符，在程序中使用依赖关系描述符可以直接使用Spring框架自动配置的一站式服务。如果想创建一个Spring Web应用程序，只需添加spring-boot-starter-web依赖项，默认情况下，这将拉出所有常用的库来开发Spring MVC应用程序，如spring-webmvc，jackson-json ，validation-api和Tomcat。换句话说，Spring Boot收集所有公共依赖项并在一个地方定义它们，并允许开发人员使用它们，而不是每次创建新应用程序时重新发明轮子。因此，pom.xml变得比传统Spring应用程序使用的小得多。

Springboot相比较与原始的SpringMVC框架，另一大特点就是自动配置。在Springboot配置文件中配置依赖项就会为应用程序自动配置，例如，在配置文件插入spring-boot-starter-web。Spring Boot会自动配置常见的Bean，如DispatcherServlet，ResourceHandlers，MessageSource。此外，如果使用的是spring-boot-starter-jdbc，Spring Boot会自动注册DataSource，EntityManagerFactory和TransactionManager bean，并从application.properties文件中读取连接详细信息。

默认情况下，在Springboot用用程序项目在创建时就会内置一个tomcat服务器，为开发者减少了配置服务器的操作，为开发部署节省了时间，提高了效率。开发人员无需担心设置servlet容器并在其上部署应用程序。应用程序可以使用其嵌入式服务器作为可运行的jar文件自行运行。如果需要使用单独的HTTP服务器，则只需要排除默认依赖项，Spring Boot为HTTP服务器提供单独的启动程序，以帮助使此过程尽可能简单。使用嵌入式服务器创建独立的Web应用程序不仅便于开发，而且是企业级应用程序的合法解决方案，并且在微服务领域越来越有用。能够将整个服务（例如，用户身份验证）包装在暴露API的独立且完全可部署的工件中，使得分发和部署更加快速和易于管理。

## 2.3mybatis

MyBatis数据映射器框架使关系数据库与面向对象的应用程序的使用变得更加容易。 MyBatis使用XML描述符将对象与存储过程或SQL语句耦合在一起。 简单性是MyBatis数据映射器相对于对象关系映射工具的最大优势。要使用MyBatis数据映射器，您需要依赖自己的对象，XML和SQL。 没有什么可以学到的，你还不知道。 使用MyBatis Data Mapper，可以轻松获得SQL和存储过程的全部功能。

## 2.4shiro

Apache Shiro的首要目标是易于使用和理解。

以下是Apache Shiro可以做的一些事情：

验证用户以验证其身份

为用户执行访问控制，例如：

确定是否为用户分配了某个安全角色

确定是否允许用户执行某些操作

在任何环境中使用Session API，即使没有Web容器或EJB容器也是如此。

在身份验证，访问控制或会话生命周期内对事件做出反应。

聚合用户安全数据的1个或多个数据源，并将其全部显示为单个复合用户“视图”。

启用单点登录（SSO）功能

无需登录即可为用户关联启用“记住我”服务

...

还有更多 - 所有这些都集成在一个易于使用的内聚API中。

Shiro尝试为所有应用程序环境实现这些目标 - 从最简单的命令行应用程序到最大的企业应用程序，而不会强制依赖其他第三方框架，容器或应用程序服务器。当然，该项目旨在尽可能地融入这些环境，但它可以在任何环境中开箱即用

Apache Shiro是一个具有许多功能的综合应用程序安全框架。 下图显示了Shiro关注其能量的位置，此参考手册将以类似方式组织：

Shiro针对Shiro开发团队所称的“应用程序安全的四大基石” - 身份验证，授权，会话管理和加密：

身份验证：有时称为“登录”，这是证明用户是他们所说的人的行为。

授权：访问控制的过程，即确定“谁”可以访问“什么”。

会话管理：即使在非Web或EJB应用程序中，也可以管理特定于用户的会话。

密码学：使用加密算法保持数据安全，同时仍然易于使用。

还有其他功能可以在不同的应用程序环境中支持和强化这些问题，尤其是：

Web支持：Shiro的Web支持API可帮助轻松保护Web应用程序。

缓存：缓存是Apache Shiro API中的第一层公民，可确保安全操作保持快速高效。

并发：Apache Shiro支持具有并发功能的多线程应用程序。

测试：存在测试支持以帮助您编写单元和集成测试，并确保您的代码按预期受到保护。

“运行方式”：允许用户假定其他用户的身份（如果允许）的功能，有时在管理方案中很有用。

“记住我”：记住用户在会话中的身份，因此他们只需要在必要时登录。

# 4 系统模块设计

## 4．1系统功能模块

本模块的设计主要目的是把本文要设计的论坛的相关功能模块化，把要实现的功能拆分或组合成一个模块。模块与模块之间相互独立又保持一定的相关性，模块与模块之前是不可分割的关系，一个设计优良的系统就是由不同的模块的协调运作实现完美运行的。

本论坛主要做了两大部分，分别是客户端和审核端，这两个部分下属又包括了很多模块。客户端主要完成与用户相关的一些操作，比如注册及登录功能、个人信息维护功能、悬赏功能、回答悬赏功能、刷一刷、答案排行等。客户端要实现的功能模块如图4.1所示

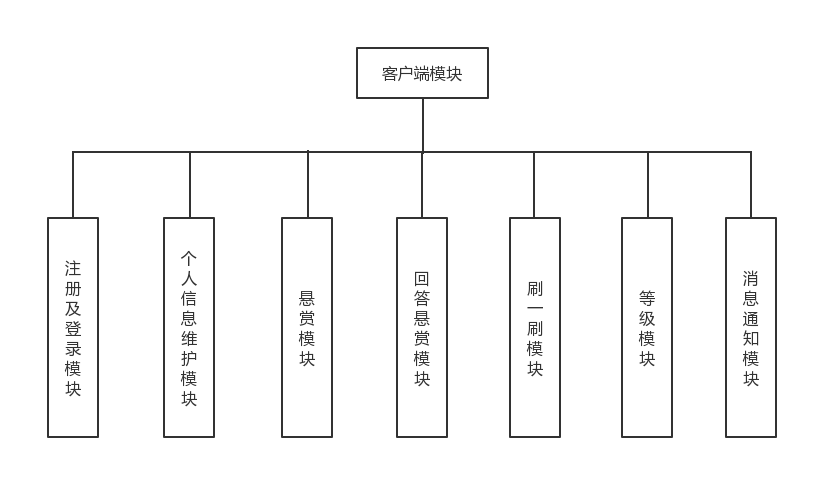


图4.1 客户端功能模块图

审核端主要完成管理员的一些操作，比如登录功能、审核图片悬赏答案、审核视频悬赏答案。审核端要实现的功能模块如图4.2所示。

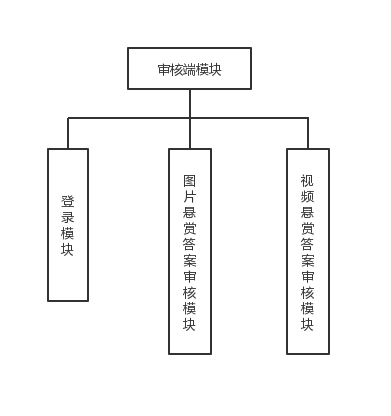


图4.1 审核端功能模块图

## 4.2 整体实现架构

千里眼在线图片、视频悬赏论坛分为客户端模块和审核端模块，是B/S架构的系统，采用了数据库持久化存储数据。开发架构如图4.3所示。

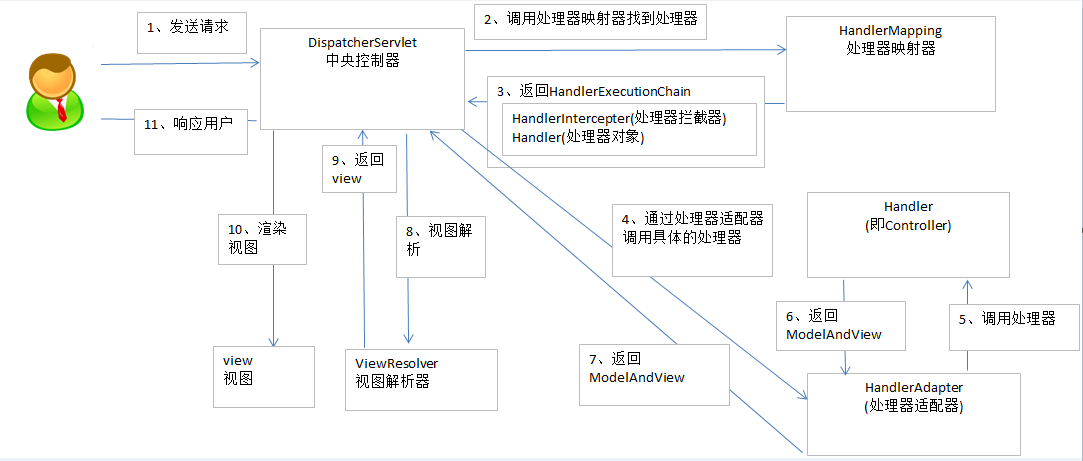


图4.3 开发架构图

## 5.1 MySQL数据库介绍

数据库介绍+范式基础+如何实现高效的数据库设计

数据库是一个存储数据集合的独立应用程序。 每个数据库都有一个或多个不同的API，用于创建，访问，管理，搜索和复制它所拥有的数据。也可以使用其他类型的数据存储，例如文件系统上的文件或内存中的大型哈希表，但是对于这些类型的系统，数据获取和写入不会那么快速和容易。

如今，我们使用关系数据库管理系统（RDBMS）来存储和管理大量数据。 这称为关系数据库，因为所有数据都存储在不同的表中，并且使用主键或称为外键的其他键建立关系。

关系数据库管理系统（RDBMS）是一种软件使您能够实现包含表，列和索引的数据库、保证各种表行之间的参照完整性、自动更新索引、解释SQL查询并组合来自各种表的信息。

MySQL是一种快速，易于使用的RDBMS，可用于许多小型和大型企业。 MySQL由瑞典公司MySQL AB开发，营销和支持。由于很多很好的理由，MySQL变得如此受欢迎MySQL是在开源许可下发布的。因此无需支付任何费用。

MySQL本身就是一个非常强大的程序。它处理最昂贵和功能最强大的数据库包的大部分功能。MySQL使用着名的SQL数据语言的标准形式。MySQL可以在许多操作系统上运行，并且可以使用许多语言，包括PHP，PERL，C，C ++，JAVA等。MySQL工作非常快，即使对于大型数据集也能正常工作。MySQL对PHP非常友好，PHP是用于Web开发的最受欢迎的语言。MySQL支持大型数据库，表中包含多达5000万行或更多行。表的默认文件大小限制为4GB，但您可以将此值（如果您的操作系统可以处理它）增加到理论上的800万TB（TB）。MySQL是可定制的。开源GPL许可证允许程序员修改MySQL软件以适应他们自己的特定环境。

## 5.2 NOSQL及Redis数据库介绍

几乎每个应用程序都必须存储数据，并且通常需要大量快速变化的数据。直到最近，大多数应用程序使用关系数据库管理系统（简称RDBMS）存储数据，如Oracle，MySQL或PostgreSQL。然而，最近，由于需要以更有效的方式存储无模式数据NoSQL，因此出现了一种新的数据存储模式。选择是使用SQL还是NoSQL通常是设计成功应用程序的重要第一步。在选择是否使用SQL或NoSQL来存储数据时，有两件重要的事情要考虑：它的性质和使用模式。某些数据非常适合关系存储引擎，而其他数据则受益于Nois引擎（如Redis或其替代品）的无模式特性。

在Web上工作时，数据和数据模型可能会随着增加的功能和业务更新而不断变化。在关系数据库中发展模式以支持这些更改是一个痛苦的过程， 特别适合非关联存储的数据示例包括事务详细信息，历史数据和服务器日志。这些通常是高度动态的，经常变化，并且它们的存储趋于快速增长，进一步加剧了调整模式以存储它们的问题。它们通常也不会感觉到“关系” 。也就是说，它们中的数据不会在与其他类型数据的关系中散开。这是一个很好的迹象，表明他们可以使用RDBMS以外的东西。

衡量NoSQL适用性的另一种方法是查看是否因性能原因而发现自己的数据非规范化，并且不再受益于关系系统的某些优势，例如一致性和冗余检查。NoSQL数据库通常不提供ACID（原子性，一致性，隔离性，持久性），或者仅部分地执行它。这允许他们做出一些其他情况下无法实现的权衡。 Redis通过设计提供部分ACID合规性，因为它是单线程（保证一致性和隔离），并且如果始终配置appendfsync，则提供完全合规性，同时提供持久性。表现也可能是一个关键因素。 NoSQL数据库通常更快，特别是对于写操作，使它们非常适合大量写入的应用程序。

Redis是一个开源（BSD许可），内存数据结构存储，用作数据库，缓存和消息代理。它支持数据结构，如字符串，散列，列表，集合，带有范围查询的排序集，位图，超级日志，具有半径查询和流的地理空间索引。 Redis具有内置复制，Lua脚本，LRU驱逐，事务和不同级别的磁盘持久性，并通过Redis Sentinel提供高可用性并使用Redis Cluster自动分区。

您可以对这些类型运行原子操作，例如附加到字符串;递增哈希值;将元素推送到列表中;计算集合交集，并集和差异;或者在排序集中获得排名最高的成员。

为了实现其出色的性能，Redis使用内存数据集。根据您的使用情况，您可以通过每隔一段时间将数据集转储到磁盘或通过将每个命令附加到日志来保留它。如果您只需要功能丰富的网络内存缓存，则可以选择禁用持久性。

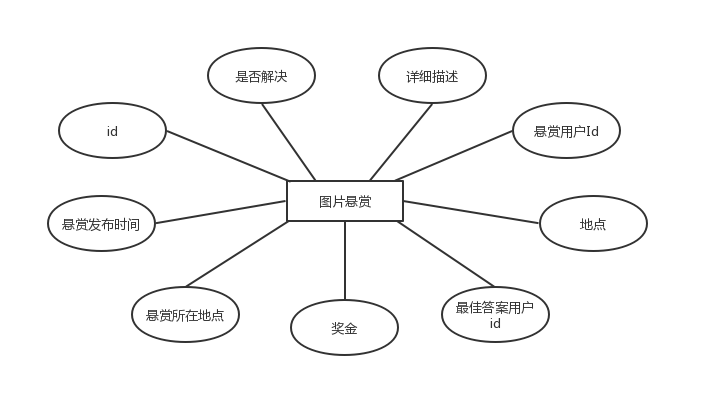
Redis还支持简单到设置的主从异步复制，具有非常快速的非阻塞第一次同步，自动重新连接以及在网络分割上的部分重新同步。其他功能包括：交易、发布/订阅、Lua脚本、钥匙的生存时间有限、LRU逐出钥匙、自动故障转移

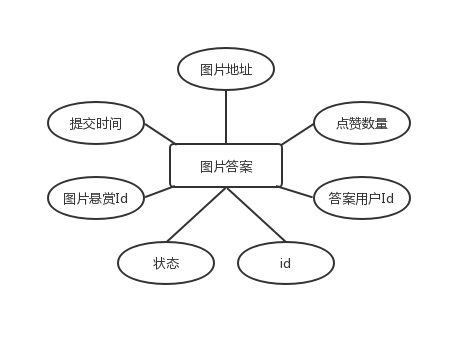
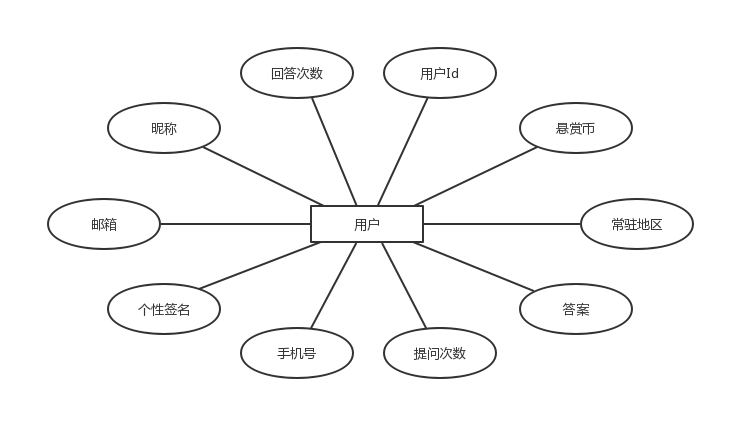
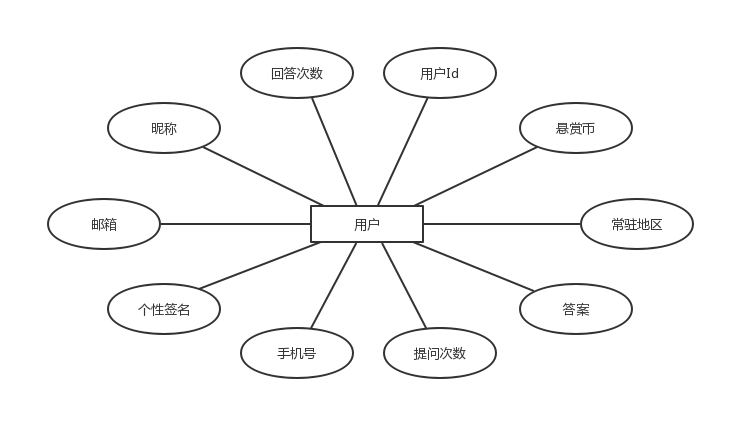
Redis是用ANSI C编写的，适用于大多数POSIX系统，如Linux，\* BSD，OS X，没有外部依赖性。 Linux和OS X是Redis开发和测试的两个操作系统，我们建议使用Linux进行部署。 Redis可能在Solaris衍生系统（如SmartOS）中工作，但支持是最好的努力。 Windows版本没有官方支持，但Microsoft开发并维护了Redis的Win-64端口。

所有这些都说，尽管NoSQL感觉更灵活，但在RDBMS中存储关系数据也有很好的论据。如果您具有非常适合规范化的可预测数据，则可以获得使用关系数据存储引擎的好处。在做出决定之前，请始终查看数据。像Redis这样的NoSQL数据库速度快，容易扩展，非常适合许多现代问题。一旦加权了所有选项，在SQL（对于稳定的，可预测的，关系数据）和NoSQL（对于临时的，高度动态的数据）之间进行选择应该是一件容易的事。

## 5.3 数据库结构设计

本论坛中分为人员信息分为普通用户和悬赏审核管理员。普通用户和悬赏审核管理员在实际操作中具有的权限不同。普通用户具有增加、修改、删除悬赏信息、答案信息、个人信息的权限。管理员具有修改答案状态的权限。本文绘制了Mysql数据库的相应E-R图。





## 5.4 数据库表设计

数据库的设计原则上要按照第三范式进行设计，一边进行后续的拓展使用，本论坛在数据库设计过程中本着第三范式的原则，在一定程度上提高了整个数据库应用的整体性能。

在本章节中我们会将设计的数据库表进行展示。

1、用户信息表

用户信息表记录了每个普通用户的基本信息，用户表user\_mes相关字段的定义见表5.1。

表 5.1 用户表user\_mes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段说明 | 字段名 | 属性 | 备注 | |
| 用户id | user\_id | varchar(32) | 主键 |
| 邮箱 | email | varchar(64) |  |
| 个性签名 | signature | text |  |
| 手机 | phone | varchar(16) |  |
| 悬赏币 | money | decimal |  |
| 常驻地区 | region | varchar(64) |  |
| 提问个数 | ask | int |  |
| 回答个数 | answer | int |  |
| 密码 | password | varchar(32) |  |
| 昵称 | Nickname | varchar(32) |  |

2、管理员表

管理员表记录了每个管理员的id及密码信息，管理员表admin相关字段的定义见表5.2。

表 5.2 管理员表admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段说明 | 字段名 | 属性 | 备注 | |
| 管理员id | admin\_id | varchar(32) | 主键 |
| 密码 | password | varchar(32) |  |

3、账单表

账单表记录了每个账单的相关信息，账单表bill\_schedule相关字段的定义见表5.3。

表 5.3 账单表bill\_schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段说明 | 字段名 | 属性 | 备注 | |
| 账单id | bill\_id | int | 主键 |
| 金额 | money | decimal |  |
| 账单类型 | type | int |  |
| 创建账单的时间 | ctime | timestamp |  |
| 悬赏id | app\_id | int |  |

4、等级信息表

等级信息表记录了每个等级对应的起始经验和最高经验，由普通用户在线时长及答案分数表中查询计算得到普通用户的总经验值之后通过与该表比对，得到该用户的等级。等级信息表level相关字段的定义见表5.4。

表 5.4 等级信息表level

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段说明 | 字段名 | 属性 | 备注 | |
| 等级 | level | int | 主键 |
| 起始经验 | start\_score | int |  |
| 最高经验 | end\_score | int |  |

5、普通用户在线时长及答案分数表

普通用户在线时长及答案分数表记录了每个普通用户的在线时长及回答悬赏所得到的分数，此表内的信息用于计算该用户的等级。用户表level\_score相关字段的定义见表5.1。

表 5.1 消息表level\_score

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段说明 | 字段名 | 属性 | 备注 | |
| id | id | int | 主键 |
| 用户id | user\_id | varchar32) |  |
| 在线时长 | login\_time | int |  |
| 回答悬赏得分 | answer\_score | int |  |

6、消息表

消息表记录了要提示给用户的相关消息，其中包括回答悬赏提示、答案审核通过通知、答案中标通知、答案涉嫌违规通知、充值通知、发布悬赏通知等。账单表message相关字段的定义见表5.6。

表 5.6 用户表message

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段说明 | 字段名 | 属性 | 备注 | |
| id | id | int | 主键 |
| 用户id | user\_id | varchar32) |  |
| 消息内容 | comment | text |  |
| 消息发布时间 | time | timestamp |  |
| 消息类型 | type | int |  |
| 消息详情 | link | varchar(128) |  |
| 消息阅读状态 | state | int |  |

7、图片答案表

图片答案表包含了回复图片悬赏的答案的相关信息。图片答案表pic\_answer相关字段的定义见表5.7。

表 5.7 图片答案表pic\_answer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段说明 | 字段名 | 属性 | 备注 | |
| id | id | int | 主键 |
| 用户id | user\_id | varchar32) |  |
| 图片悬赏id | picapp\_id | int |  |
| 上传图片地址 | pic\_adress | varchar(256) |  |
| 回答时间 | answer\_time | time\_stamp |  |
| 答案状态 | state | int |  |
| 点赞量 | awesome | int |  |

7、图片悬赏表

图片悬赏含了回复图片悬赏的相关信息。图片悬赏表pic\_app\_mes相关字段的定义见表5.7。

表 5.7 图片答案表pic\_answer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段说明 | 字段名 | 属性 | 备注 | |
| id | id | int | 主键 |
| 用户id | user\_id | varchar32) |  |
| 悬赏城市 | city | varchar(64) |  |
| 悬赏简介 | introduction | text |  |
| 悬赏发布时间 | app\_time | time\_stamp |  |
| 悬赏金额 | money | decimal |  |
| 是否解决 | solve | int |  |
| 正确答案用户id | right\_user\_id | varchar(32) |  |
| 所属地区 | region | varchar(64) |  |

8、图片点赞表

图片点赞表包含了图片悬赏答案点赞的相关信息。图片点赞表pic\_awsome相关字段的定义见表5.7。

表 5.8 图片点赞表pic\_awsome

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段说明 | 字段名 | 属性 | 备注 | |
| id | id | int | 主键 |
| 悬赏答案id | answeri\_id | int |  |
| 点赞用户id | user\_id | varchar(32) |  |

9、在线时间统计表

图片统计表包含了回复用户本次登录的在线时间的相关信息，用于计算每次用户登录的在线时长。在线时间统计表time\_satistic相关字段的定义见表5.7。

表 5.9 在线时间统计表time\_satistic

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段说明 | 字段名 | 属性 | 备注 | |
| id | id | int | 主键 |
| 用户id | user\_id | varchar32) |  |
| 登录时间 | login\_time | timestamp |  |
| 最后操作时间 | latestOptime | timestamp |  |

10、视频答案表

视频答案表包含了回复视频悬赏的答案的相关信息。图片答案表video\_answer相关字段的定义见表5.7。

表 5.10 视频答案表video\_answer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段说明 | 字段名 | 属性 | 备注 | |
| id | id | int | 主键 |
| 用户id | user\_id | varchar32) |  |
| 视频悬赏id | videoapp\_id | int |  |
| 上传图片地址 | video\_adress | varchar(256) |  |
| 回答时间 | answer\_time | time\_stamp |  |
| 答案状态 | state | int |  |
| 点赞量 | awesome | int |  |

11、视频悬赏表

视频悬赏含了回复视频悬赏的答案的相关信息。视频悬赏表video\_app\_mes相关字段的定义见表5.7。

表 5.11 视频答案表video\_answer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段说明 | 字段名 | 属性 | 备注 | |
| id | id | int | 主键 |
| 用户id | user\_id | varchar32) |  |
| 悬赏城市 | city | varchar(64) |  |
| 悬赏简介 | introduction | text |  |
| 悬赏发布时间 | app\_time | time\_stamp |  |
| 悬赏金额 | money | decimal |  |
| 是否解决 | solve | int |  |
| 正确答案用户id | right\_user\_id | varchar(32) |  |
| 所属地区 | region | varchar(64) |  |

12、视频点赞表

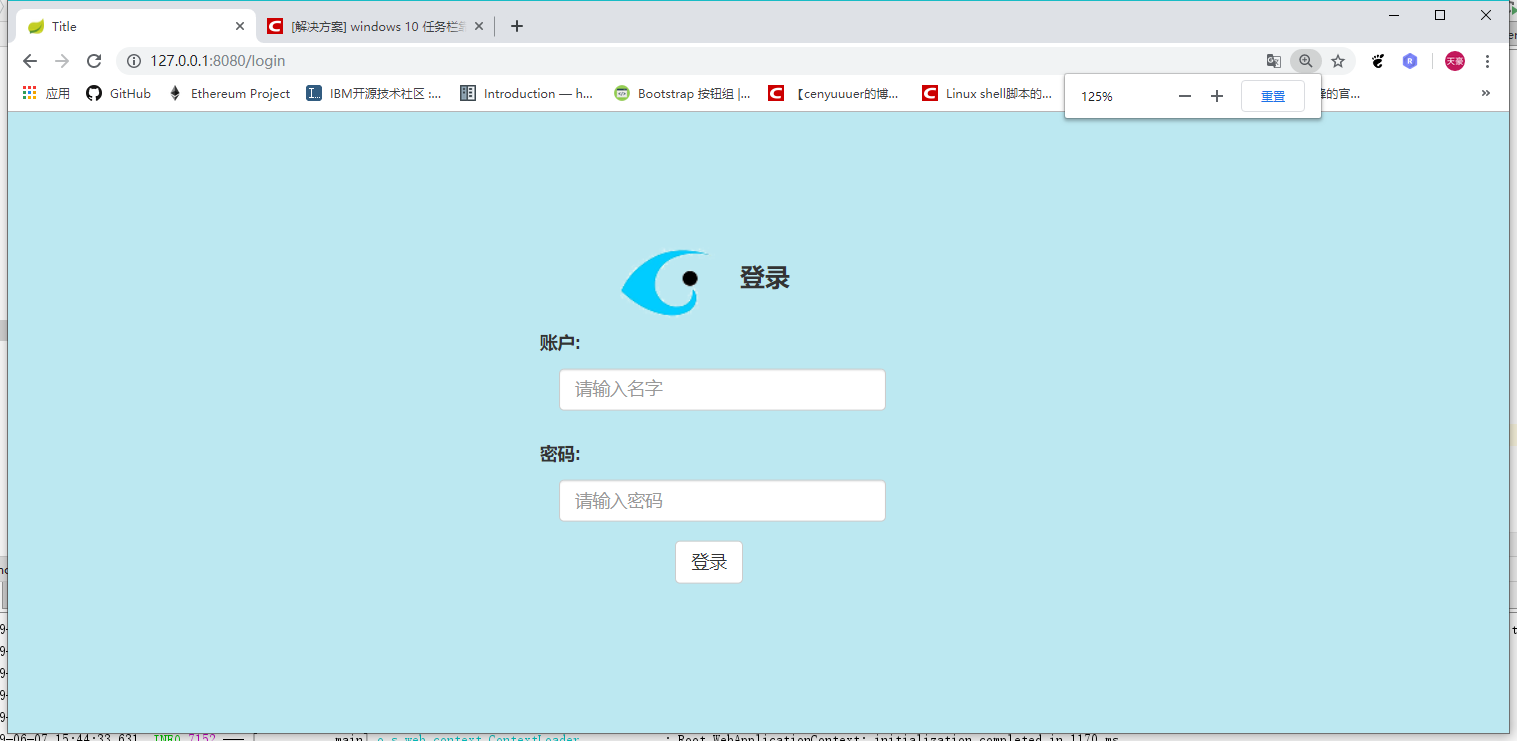
视频点赞表包含了视频悬赏答案点赞的相关信息。视频点赞表video\_awsome相关字段的定义见表5.7。

表 5.8 图片点赞表video\_awsome

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段说明 | 字段名 | 属性 | 备注 | |
| id | id | int | 主键 |
| 悬赏答案id | answeri\_id | int |  |
| 点赞用户id | user\_id | varchar(32) |  |

## 6.1登录及注册模块设计

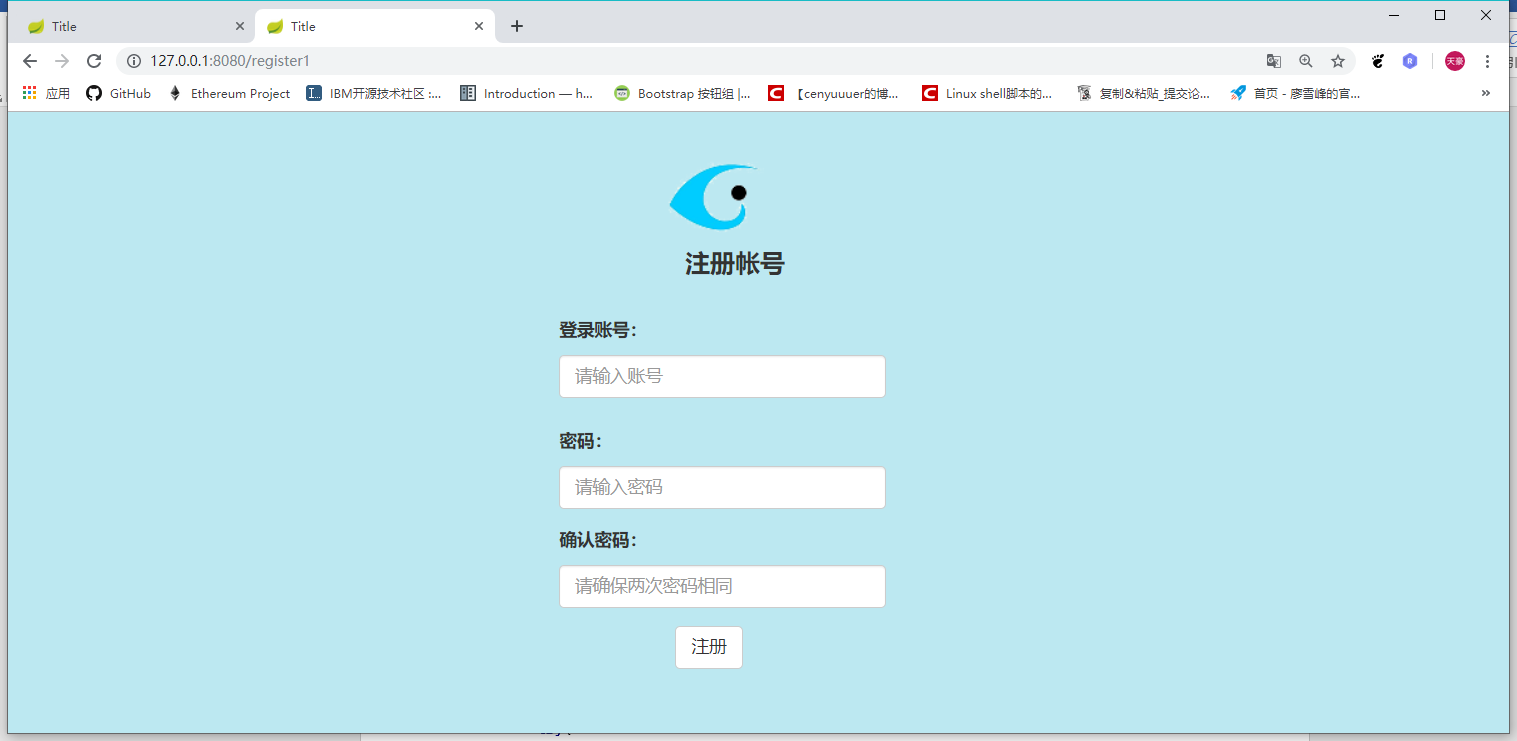
在登录模块设计中，用户根据用户名、密码登录系统。本系统分别为用户和管理员设置了不同的登录界面，利用html和css绘制界面的主题，组件库使用的是基于jquery的bootstrap，页面解释器使用的是thymeleaf。使用javascript对验证表单的元素进行基本的验证，不满足正确格式的用户名会给出提示，如果元素的的内容格式完全正确才会将数据提交至后台。处于用户信息安全的信息考虑，在注册登录过程中会使用shiro安全框架对用户的密码进行加密，加密方法使用md5加密方式，加salt迭代十次得到最终密码加密字符串，后台用此字符串与数据库中的字符串进行比对，若一样则登陆成功过，若不一样则赶回错误提示。登陆页面如图6.1所示。



主要程序代码：

**public int** checkPass(String id, String pwd) {  
 Md5Hash md5Hash = **new** Md5Hash(pwd,**"xth.com"**,10);  
 pwd = md5Hash.toString();  
 UsernamePasswordToken token = **new** UsernamePasswordToken(id,pwd);  
 DefaultSecurityManager securityManager = **new** DefaultSecurityManager();  
 AdminRealm adminRealm = **new** AdminRealm();  
 adminRealm.setPwd(**userMesMapper**.selectByPrimaryKey(id).getPassword());  
 securityManager.setRealm(adminRealm);  
 SecurityUtils.*setSecurityManager*(securityManager);  
 Subject subject = SecurityUtils.*getSubject*();  
 **try** {  
 subject.login(token);  
 **if** (subject.isAuthenticated()) {  
 **return** 1;  
 }  
 }**catch** (AuthenticationException e){  
 **return** 0;  
 }  
 **return** 0;  
}

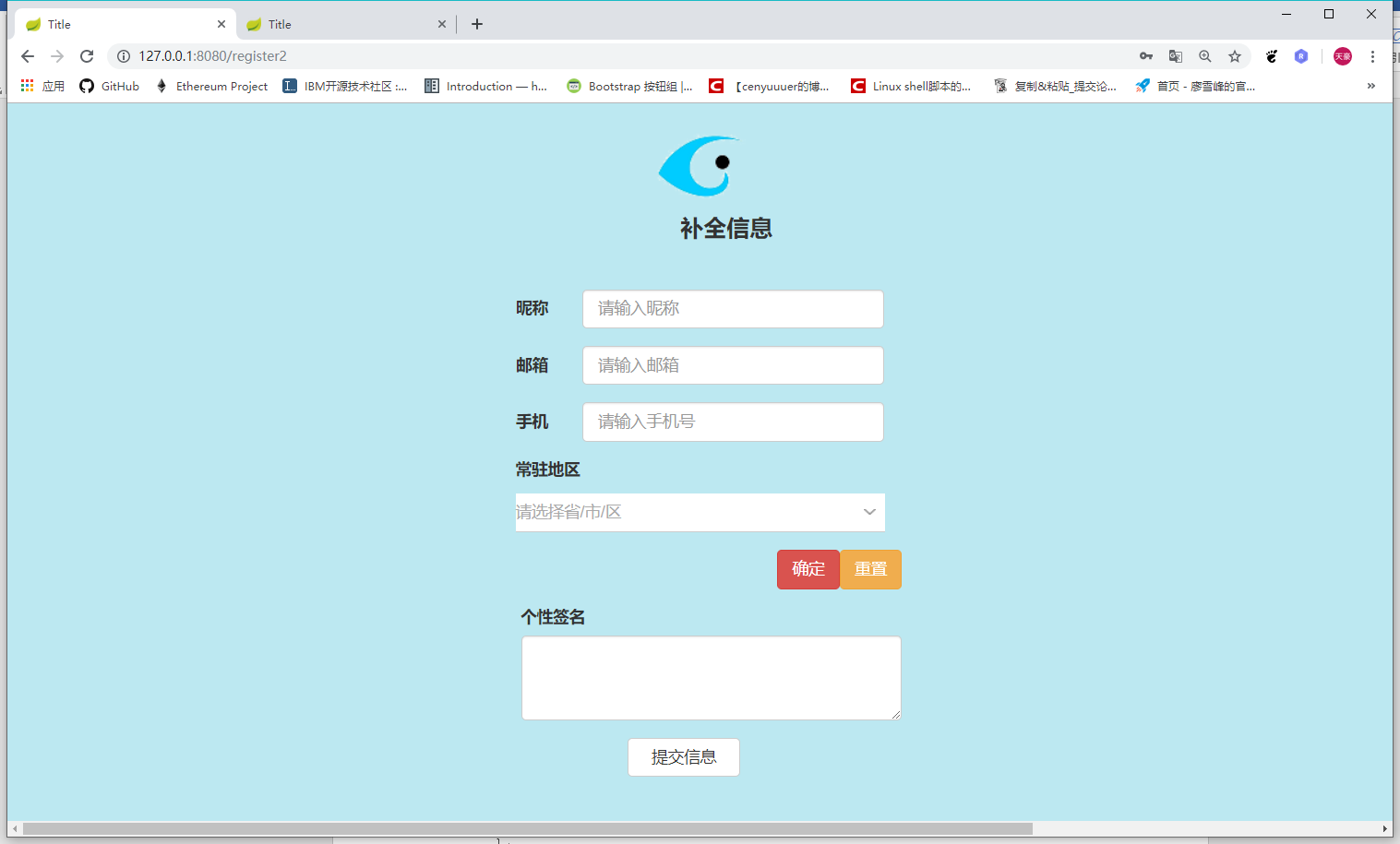
在注册模块中使用javascript对用户输入的两次密码进行一致性校验。如果两次密码输入的是相同的，用户注册所使用的账号密码信息会传至后端，后端会检验账号是否已经注册过，若已经注册过，则会返回账号已经注册过的消息提示。若前后端审核都通过，那么会跳转到输入个人基本信息的界面。注册账号的页面如图6.2所示。



注册逻辑代码：

**public int** insertUser(String userId,String password){  
  
 Md5Hash md5Hash = **new** Md5Hash(password,**"xth.com"**,10);  
 password = md5Hash.toString();  
 UserMes um = **new** UserMes();  
 um.setUserId(userId);  
 um.setPassword(password);  
 **try**{  
 **userMesMapper**.insert(um);  
 **levelScoreMapper**.insert(userId);  
 }**catch** (Exception e){  
 System.***out***.println(e.getMessage());  
 **return** 0;  
 }  
 **return** 1;  
}

在注册模块的输入个人信息模块中，需要用户输入昵称、邮箱、手机、常驻地区、以及个性签名。其中常驻地区用于在用户访问悬赏大厅的时候职能的给用户调转到其常驻地区的悬赏信息页面。在本页面中所有信息都必须填写，使用javascript进行信息格式的校验，若数据无误则传入后端。注册个人信息页面如图6.3所示。

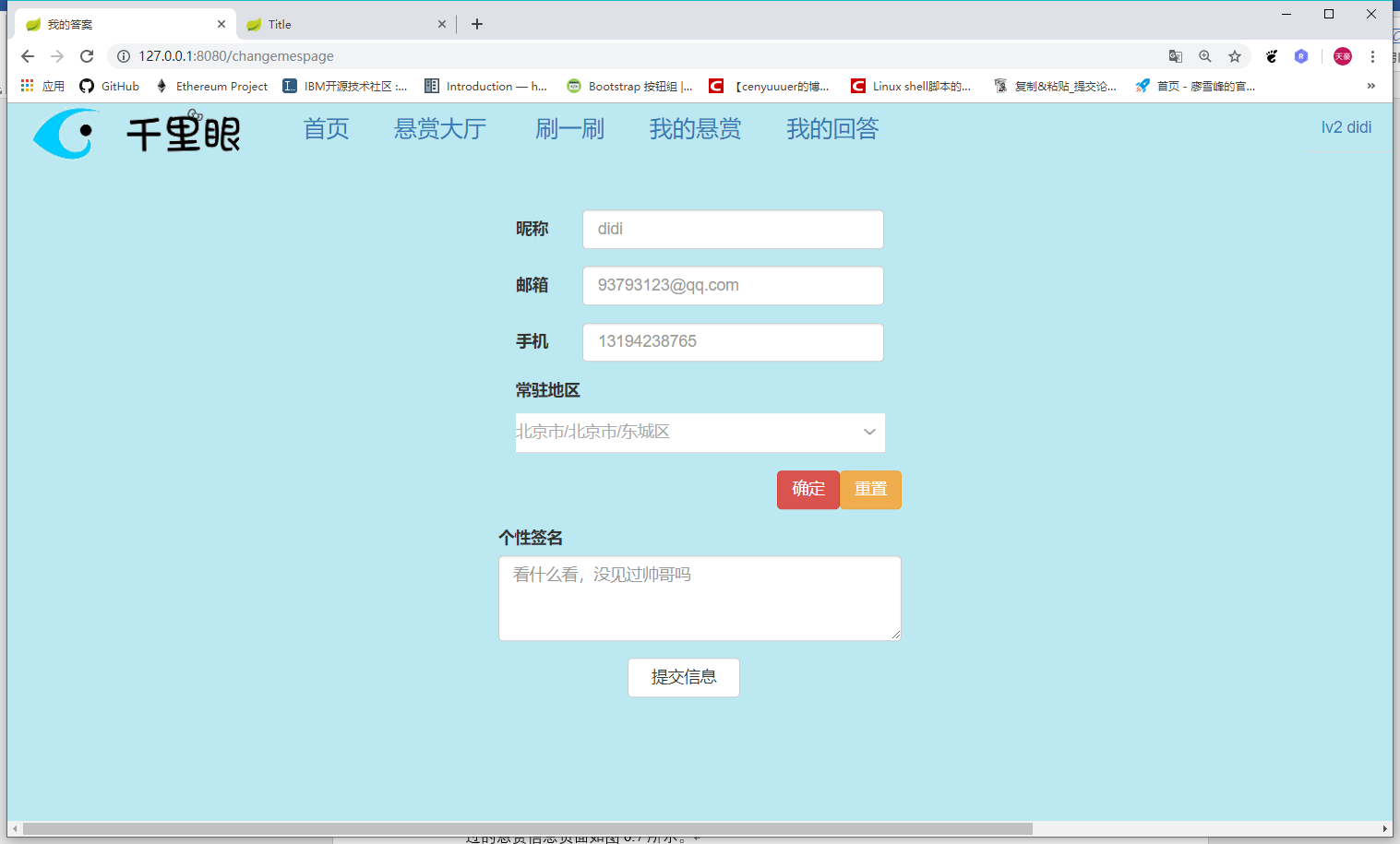


主要代码：

**public** String register2(HttpServletRequest request,HttpServletResponse response) **throws** IOException {  
 HttpSession session = request.getSession();  
 String userId = (String) session.getAttribute(**"userId"**);  
 String nickName = (String)request.getParameter(**"nickName"**);  
 String email = (String)request.getParameter(**"email"**);  
 String phone = (String ) request.getParameter(**"phone"**);  
 String region = (String) request.getParameter(**"city-picker3"**);  
 String signature = request.getParameter(**"signature"**);  
 session.setAttribute(**"nickName"**,nickName);  
 UserMes um = **new** UserMes();  
 um.setUserId(userId);  
 um.setNickname(nickName);  
 um.setEmail(email);  
 um.setPhone(phone);  
 um.setRegion(region);  
 um.setSignature(signature);  
  
 **if**(**userServiceimpl**.updateMes(um)==1)  
 **return "index"**;  
 **else** {  
 response.setContentType(**"text/html;charset=utf-8"**);  
 PrintWriter out = response.getWriter();  
 out.println (**"<script language=javascript>alert('注册失败，请稍后再试')</script>"**);  
 }  
 **return "/"**;  
}

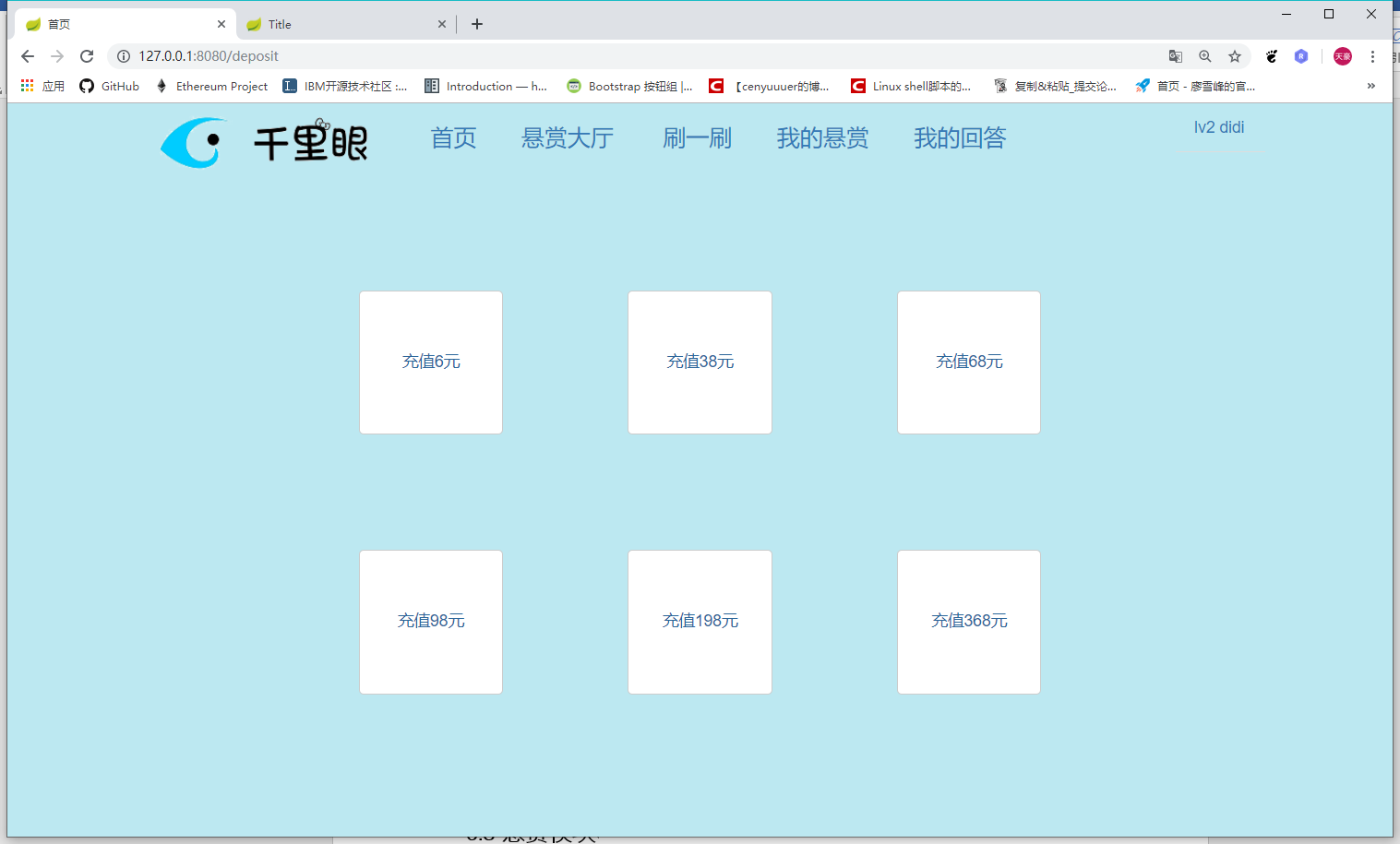
## 6．2个人信息维护模块

在个人信息维护模块中，用户可以维护修改自己的个人信息，包括密码、昵称、邮箱、手机号、个性签名等信息。为保证用户仅能修改其自身的账户信息及密码，后台会审核当前用户是否具有修改目标用户信息及密码的权限，若具有权限则允许修改，若没有权限则返回警告提示。修改个人信息模块如图6.4所示。



密码修改界面会要求用户输入两次密码，确保用户输入密码的正确性，因为在后台会审核当前用户是否具有修改目标用户的密码权限，所以这里并没有要求用户在输入一次密码，简便了操作，修改密码界面如图6.5所示。

此系统模拟了一个充值页面，用户可以在此页面点击按钮进行充值。充值页面如图6.6所示。



## 6.3悬赏模块

悬赏模块其中包括展示已经发布过的悬赏大厅、个人悬赏信息、展示此悬赏的答案、发布图片悬赏和发布视频悬赏四个子功能。

展示发布过的悬赏信息，包括具体地点、所在区域、当前悬赏状态、悬赏奖金、发布时间以及可选择查看答案以及删除悬赏的操作。悬赏状态包括已解决和未解决。查看答案会弹出回答该悬赏答案的页面，若发布该悬赏的用户已经在所有回答中选择了中标答案，则此页面将仅作于展示悬赏答案使用，若并未选择中标答案，则可在此页面选择一个答案为最终中标答案。删除悬赏操作会以弹窗的方式让用户再次确认删除该悬赏，以防止误删。展示发布过的悬赏信息页面如图6.7所示。



在展示个人悬赏信息页面，若有其他用户回答了此悬赏，发布悬赏的用户可以在对应悬赏的条目下看到一个查看回答按钮，点击查看回答可以跳转到回答界面。若用户并未为此悬赏选择最优答案，则在每个答案提交人昵称旁边会生成一个中标按钮，用户可以选择一个最优答案。若已经选择了最优答案，则在最优答案提交人的昵称旁生成一个中标的标签。同时针对多人回复一个悬赏的情况，本论坛在前端展示进行了分页处理，每个用户的回答自成一页。查看答案页面如图6.8所示



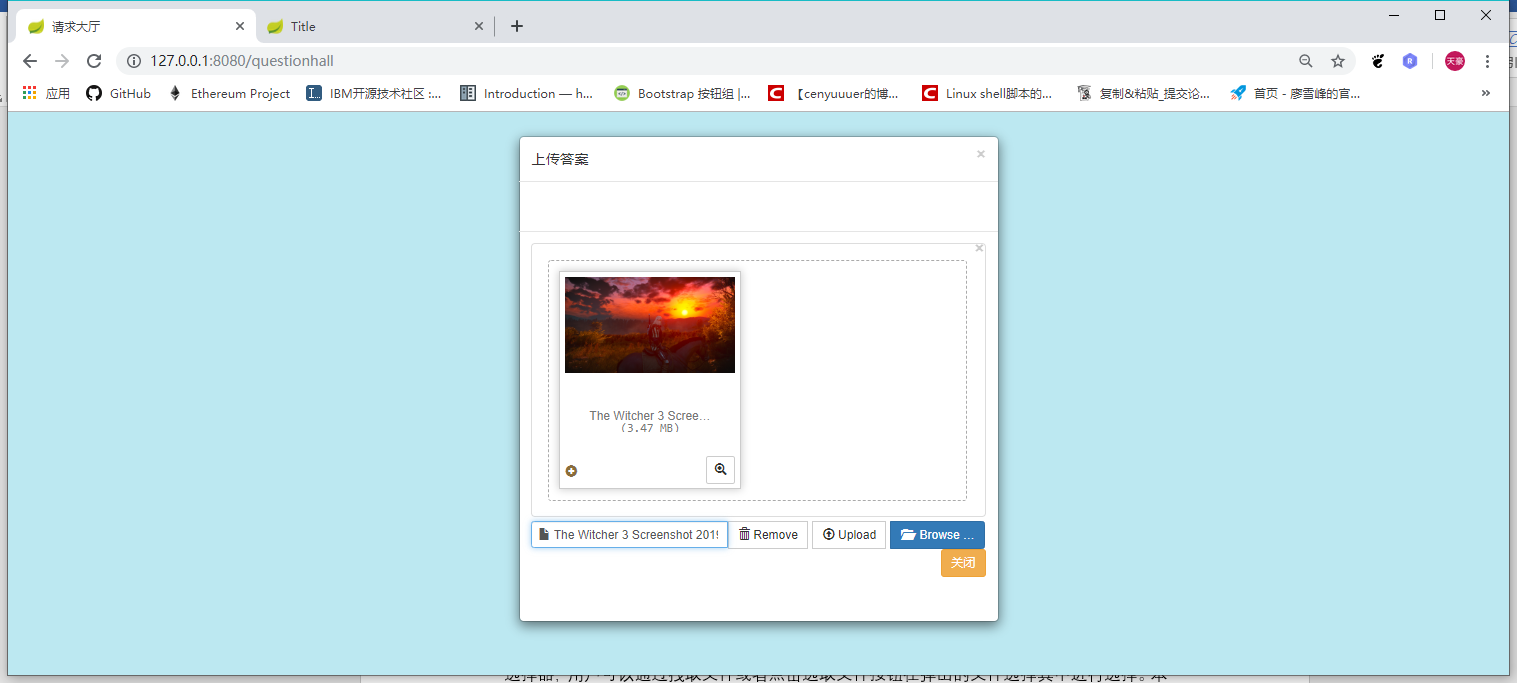
悬赏大厅，用户可以在此页面中浏览某一地区的未解决的悬赏，展示的条目默认根据选上发布的时间进行排序。条目展示内容包括具体地点、所在区域、奖金数额、发布时间和操作。悬赏大厅页面如图6.9所示。



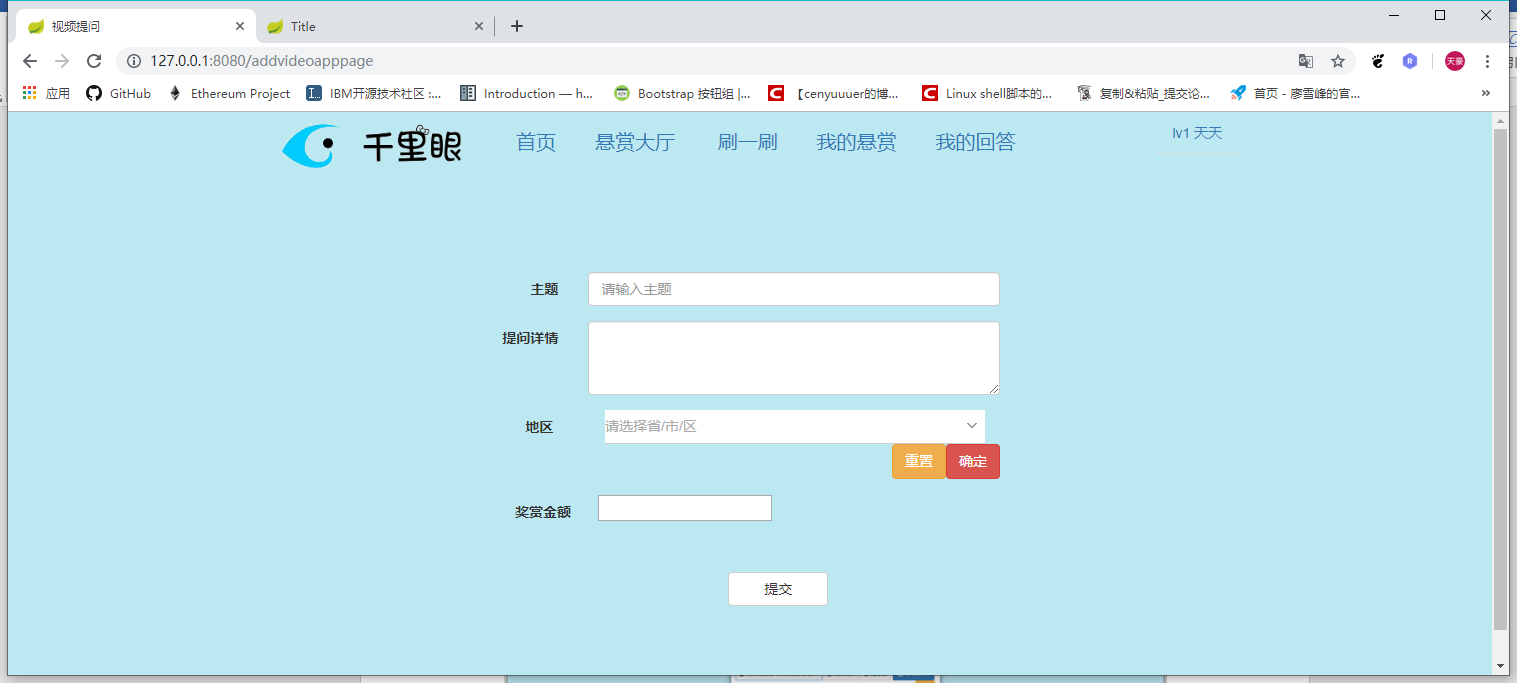
悬赏大厅中每条悬赏条目包括查看介绍以及回答操作。点击查看介绍会弹出一个bootstrap组件model模态框，在此模态框中展示发布用户填写的具体要求。查看介绍模态框如图6.10所示。



在悬赏大厅中点击回答按钮，同样会弹出一个model模态框。在此模态框中设置了文件 选择器，用户可以通过拽取文件或者点击选取文件按钮在弹出的文件选择其中进行选择。本模块中所使用的文件选择器设置为最多上传十张照片或者一个视频，文件选择器支持预览功能，用户可以点击预览按钮以查看大图或者播放视频。回答模态框如图6.11所示。



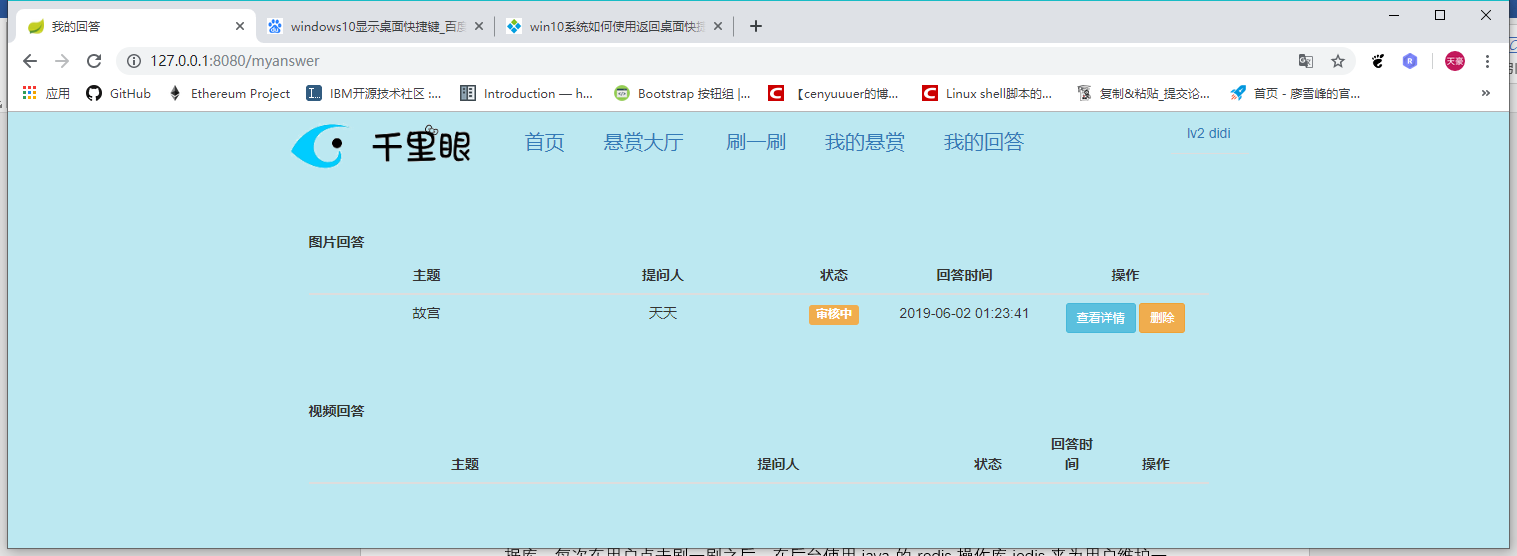
在我的悬赏页面可以发布新的悬赏。悬赏种类包括图片悬赏和视频悬赏。悬赏信息包括悬赏具体地点、悬赏地点所在地区、悬赏详细介绍、悬赏金额。添加悬赏信息页面如图6.12所示。



## 6.4答案模块

答案模块其中包括个人答案信息、最佳答案排行两个子页面。

个人答案信息页面，用户可以在此页面看到自己回复悬赏的答案信息，其中包括悬赏地点、悬赏地点所在区域、答案状态、回答时间、以及查看详情和删除亮相操作。答案状态包括审核中、已审核、中标和违规。只用已审核和中标的答案会对其他用户进行展示。状态码标签根据状态的不同会呈现不同的颜色。点击删除链接，系统会要求用户重复确认以防止用户的误操作。点击查看详情链接可以查看用户自己回答的具体内容，同时也可以看到一个悬赏下其他用户提交的答案，方便对自身的答案进行修改。回答个人信息页面如图所示。



## 6.5刷一刷模块

刷一刷模块用于随机为用户推送优秀的图片悬赏的答案和视频悬赏的答案。使用户能够在查看高质量的答案以及不断优化自己的答案内容。本模块采用的解决方案是使用redis数据库，每次在用户点击刷一刷之后，在后台使用java的redis操作库jedis来为用户维护一个属于该用户的刷一刷队列，刷到一个图片悬赏的答案或者视频悬赏的答案就向redis对应的key中追加一个答案id。队列有效时间设置为10分钟，每次用户有新的刷一刷操作都会更新队列的有效时间。为了保证给用户推送的图片悬赏答案和视频悬赏答案具有一定的随机性，存储数据库的图片悬赏答案和视频答案所用的数据结构使用的是HashSet。刷一刷页面如图6.13所示。

主要代码：

@RequestMapping(value = **"shuayishu"**)  
**public** String shuayishua(HttpServletRequest request, ModelMap modelMap, HttpServletResponse response) **throws** IOException {  
 String userId = (String) request.getSession().getAttribute(**"userId"**);  
 modelMap.addAttribute(**"level"**,level);  
 SaPicAnswer saPicAnswer = **saService**.getRandomPic(userId);   
 **if**(saPicAnswer==**null**){**return "nilshua"**;  
 }  
 **else** {  
 modelMap.addAttribute(**"saPicAnswer"**,saPicAnswer);  
 }  
 **return "picshua"**;  
}

## 6.6等级模块

等级模块提供了计算等级的功能。本论坛的等级经验值由两部分构成，分别是在线时间和回答悬赏奖励的回答分数。在线时间的统计的实现方法是在Mysql数据库中维护一个当次在线时间的统计表、等级划分经验值表、总在线时间和答案分数的汇总表，用户初次登录的时候会在表中插入数据，数据项包括本次用户id、本次登录时间、最后一次操作时间。用户每次执行操作的时候都会更新对应的最后一次操作时间。统计回答分数，用户提交的图片悬赏答案或者视频悬赏答案如果通过审核，则会获得30点回答的经验值，若答案被发布悬赏的用户选中为最优答案，则会获得200点经验值，在线时长和答题经验值有着不同的权重。每次跳转一个新页面之后后端都会重新计算当前用户的最新等级。

等级计算公式：

计算等级的主要函数如下：

**public int** getLevel(String userId){  
 LevelScore levelScore = **levelScoreMapper**.selectSumScoreByUserId(userId);  
 **int** sumScore = (**int**) (levelScore.getAnswerScore() +0.1\*levelScore.getLoginTime());  
 **int** level = **levelMapper**.selectLevel(sumScore);  
 **return** level;  
}

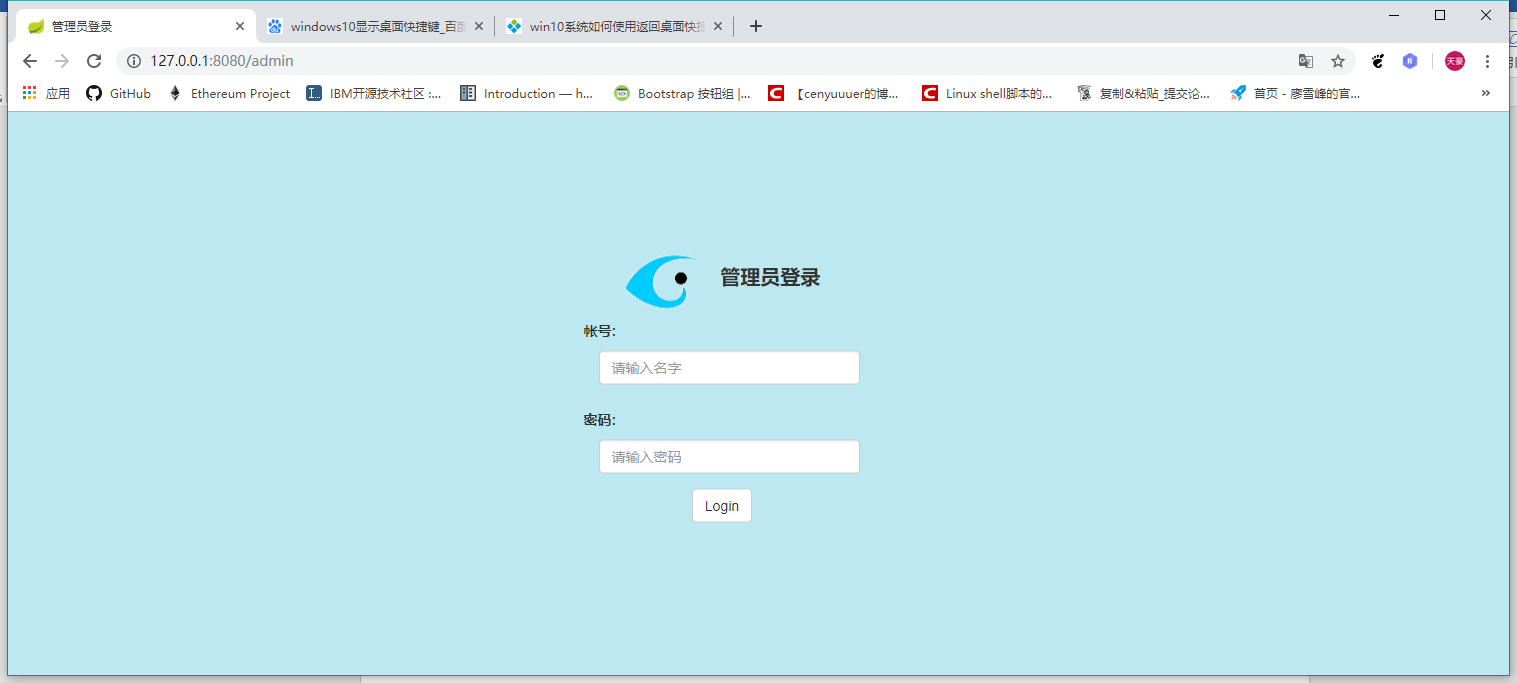
为了对用户登录时间进行及时的审核，在main方法启动前，在Springboot所有Bean注入完成后新建一个线程，该线程用于统计所有用户在当此登录的有效时长。统计当此时长代码如下：

**public void** run(){  
 **while** (**true**){  
 List<TimeSatistic>tss = **timeSatisticMapper**.selectAll();  
 **if**(tss.size()==0){*//没有账户登录* **try** {  
 *sleep*(1000);  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }**else** {*//有账户在登录* **for** (TimeSatistic ts: tss  
 ) {  
 **if**(((**new** Date().getTime()-ts.getLatestoptime().getTime()))>1000\*60\*10){  
 *//超过时间,记录时间* **timeSatisticMapper**.deleteByPrimaryKey(ts.getId());  
 String userId = ts.getUserId();  
 LevelScore levelScore = **levelScoreMapper**.selectSumScoreByUserId(userId);  
 levelScore.setLoginTime((**int**) (levelScore.getLoginTime()+(ts.getLatestoptime().getTime()-ts.getLogintime().getTime())/1000/60));  
 **levelScoreMapper**.updateByPrimaryKey(levelScore);  
 }  
 }  
 **try** {  
 *sleep*(1000);  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
}

## 6.7管理员审核模块

管理员审核模块包括管理员的登录页面、审核图片悬赏答案、审核视频悬赏答案的功能。

管理员登录模块类似于用户登录模块，管理员在前端页面输入账号和密码，后端AdminController接收到数据传给下层AdminService进行密码加密以及账号密码信息的验证。管理员登录页面如图所示。



在管理员成功登录后，会跳转至图片悬赏答案审核页面，在该页面管理员可以对用户上传的图片悬赏答案进行审核以确保用户的答案中不会出现不良的信息。此管理员模块不作为后台信息管理使用，在此不显示用户的个人信息。本页面提供给了管理员两个处理链接，一个是审核通过链接，一个是答案违规链接。点击审核通过链接后端AnswerController会调用AnswerService进行答案状态信息的更改，若更改答案状态信息发生错误，也会想前台发出一个修改错误的提示。点击违规链接，后端修改该答案的状态信息为违规。其中悬赏答案只有在审核通过的情况下才会用户登录主页参与最高点赞排行榜以及刷一刷网页。审核页面如图所示。

## 6.8通知模块（待定）