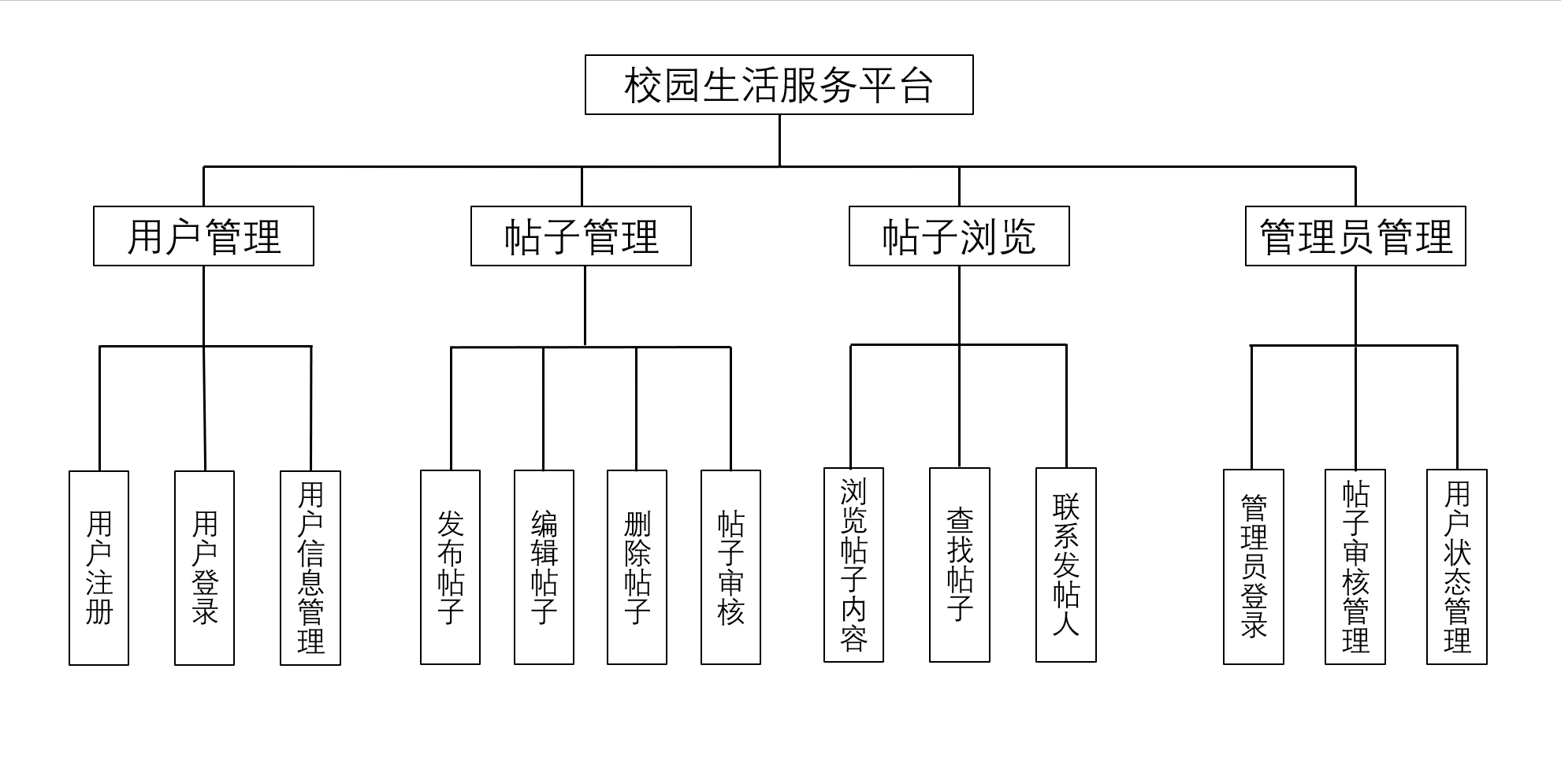
## 2.2 功能模块设计

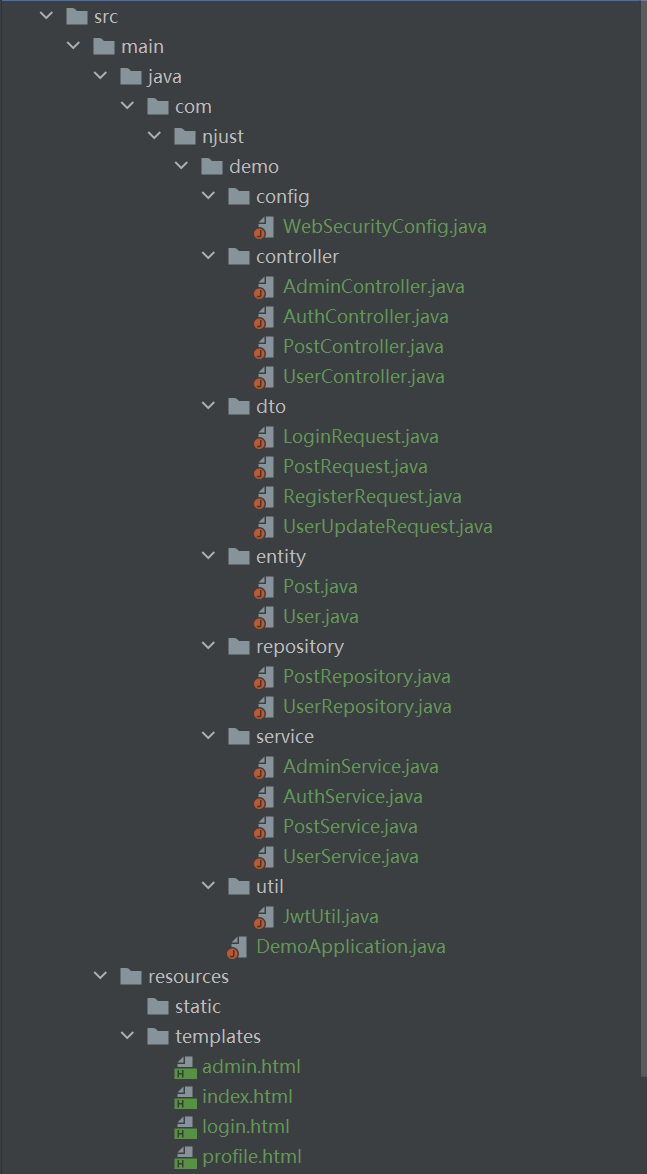
### 2.2.1 功能模块划分

基于对校园生活服务需求的全面分析，我们将系统的主要业务功能模块划分为用户管理、帖子管理、帖子浏览和管理员管理四个部分。这一划分不仅为用户提供了便捷、高效的操作体验，同时也为管理员的管理工作提供了有力支持。用户可以通过平台的不同板块发布需求或浏览其他用户发布的内容，并与发帖人取得联系，进行交流和交易。通过这样的设计，平台能够有效满足校园生活中多样化的服务需求，提升用户的整体体验和满意度。平台详细的功能模块如图x.1所示。



图x.1 平台功能模块示意图

根据图x.1功能模块示意图与Spring MVC下的项目分层架构规范，实现上述各功能点所需的各类型资源文件结构如图x.d所示



**具体文件和功能说明**

1. config 目录

* **WebSecurityConfig.java**: 配置 Spring Security 相关的内容，如认证、授权、密码加密等。

1. controller 目录

* **AuthController.java**: 处理用户认证相关的请求，如注册、登录、找回密码等。
  + register(RegisterRequest request)
  + login(LoginRequest request)
  + resetPassword(String email)
* **PostController.java**: 处理帖子相关的请求，如发布、搜索、浏览帖子等。
  + createPost(PostRequest request)
  + searchPosts(String keyword)
  + browsePosts(String category)
* **UserController.java**: 处理用户个人主页和信息修改的请求。
  + getUserProfile(Long userId)
  + updateUserProfile(UserUpdateRequest request)
  + updateAvatar(MultipartFile avatar)
* **AdminController.java**: 处理管理员相关的请求，如审核帖子、管理用户状态等。
  + reviewPost(Long postId, boolean approve)
  + manageUserStatus(Long userId, Status status)

1. dto 目录

* **LoginRequest.java**: 封装登录请求数据。
* **RegisterRequest.java**: 封装注册请求数据。
* **PostRequest.java**: 封装发布帖子请求数据。
* **UserUpdateRequest.java**: 封装用户信息更新请求数据。

1. entity 目录

* **User.java**: 用户实体类，包含用户基本信息和状态。
* **Post.java**: 帖子实体类，包含帖子内容、类别等信息。

1. repository 目录

* **UserRepository.java**: 用户数据访问接口，继承 JpaRepository。
* **PostRepository.java**: 帖子数据访问接口，继承 JpaRepository。

1. service 目录

* **AuthService.java**: 处理用户认证相关的业务逻辑。
  + registerUser(RegisterRequest request)
  + authenticateUser(LoginRequest request)
  + sendResetPasswordEmail(String email)
* **PostService.java**: 处理帖子相关的业务逻辑。
  + createPost(PostRequest request)
  + searchPosts(String keyword)
  + getPostsByCategory(String category)
* **UserService.java**: 处理用户信息相关的业务逻辑。
  + getUserById(Long userId)
  + updateUser(UserUpdateRequest request)
  + updateUserAvatar(Long userId, MultipartFile avatar)
* **AdminService.java**: 处理管理员相关的业务逻辑。
  + approvePost(Long postId, boolean approve)
  + changeUserStatus(Long userId, Status status)

1. util 目录

* **JwtUtil.java**: JWT 工具类，用于生成和验证 JWT 令牌。

1. 主应用类

* **ProjectApplication.java**: Spring Boot 主应用类，包含 main 方法。

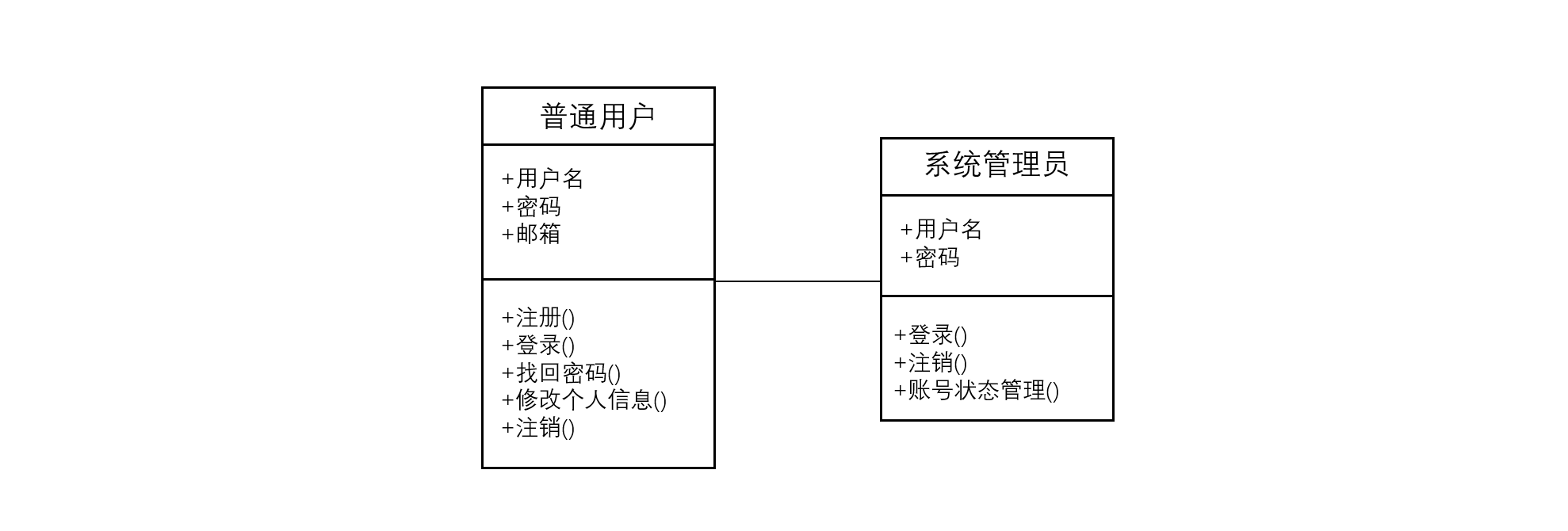
1. 资源文件目录

* **application.properties**: Spring Boot 配置文件，包含数据库、邮件服务器等配置信息。
* **templates**: Thymeleaf 模板目录，用于存放 HTML 页面。

### 2.2.2 功能模块设计

#### 2.2.2.1 用户管理模块设计

在校园生活服务平台中，用户管理模块主要负责平台用户的注册，验证，登录等一系列账号管理功能以及用户信息配置存储。用户管理类图如图x.2所示。从图中可以看出此业务功能主要服务于普通平台用户，为其提供个性化服务。



图x.2 用户管理类图

从图x.1平台功能模块示意图中可以看到，用户管理模块主要包括了用户注册，用户登录以及用户信息管理三个功能，其中用户信息管理由系统管理员负责，具体每部分功能的详细设计如下。

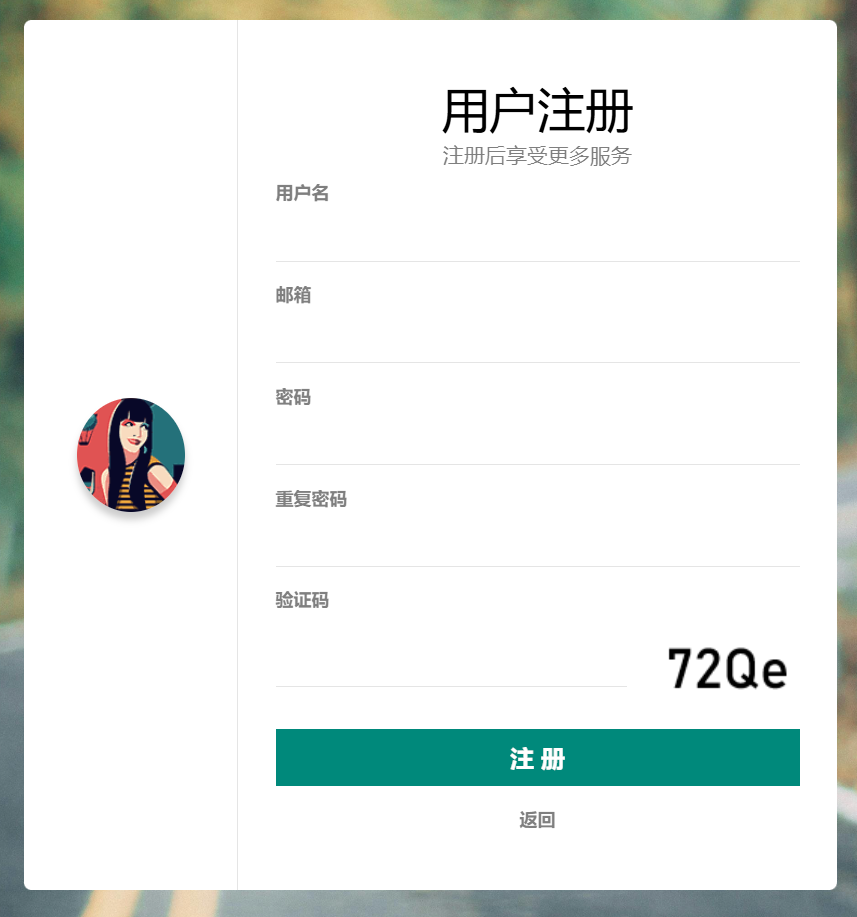
##### 1.用户注册

为了方便管理用户，为不同用户提供个性化服务。要求使用本平台的用户需要先注册个人账号。用户需要提供用户名，密码，邮箱进行注册并通过验证码验证。注册信息及详细情况如表b.1所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 输入项 | 具体要求 | 是否必填 |
| 1 | 用户名 | 只能由6-16位字母，数字组成，不可重复 | 是 |
| 2 | 密码 | 由6-12位字符，数字组成 | 是 |
| 3 | 确认密码 | 必须与“密码”项一致 | 是 |
| 4 | 邮箱 |  | 是 |
| 5 | 验证码 | 与图片提供的内容一致 | 是 |

表b.1 用户注册信息输入项设计说明表

用户注册界面UI设计如图x.3所示



图x.3 用户注册界面UI设计图

前端使用axios对用户提交的表单进行拦截与校验(用户名格式是否合法)，后端依照表b.1中的具体要求，对用户提交的注册表单进行校验，若信息不合法(用户已存在，两次密码不一致，验证码错误等)，系统将给用户相对应的提示；信息校验无误后，系统将该用户信息存入，并向用户提示注册成功，此时用户可跳转至登录界面进行账号登录。

##### 2.用户登录

已经注册本平台账号的用户在访问时可以直接输入账号密码进行登录。若用户未注册或者忘记密码，则可以根据该页面提供的按钮跳转到对应的注册页面或找回密码界面。登录信息及详细情况如表b.2所示

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 输入项 | 具体要求 | 是否必填 |
| 1 | 用户名 | 只能由6-16位字母，数字组成 | 是 |
| 2 | 密码 |  | 是 |

表b.2 用户登录信息输入项设计说明表

用户登录界面UI如图x.4所示



图x.4 用户登录界面设计图

前端使用axios对用户提交的表单进行拦截与校验，如果用户输入的登录信息不合法(用户名格式错误)，则给出用户提示，告知其非法原因。若合法则交由后端处理，对用户信息进行检验。登录成功后主动跳转到本平台首页。

若用户忘记密码，可进入找回密码界面，根据注册时提供的邮箱请求找回密码。找回密码页面输入项设计如表b.3所示

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 输入项 | 具体要求 | 是否必填 |
| 1 | 邮箱地址 |  | 是 |

表b.3 用户找回密码界面信息输入项设计说明表

用户找回密码界面UI如图x.5所示



图x.5 用户找回密码界面设计图

后端系统根据用户提交的邮箱向数据库查询是否有该用户信息，若有则返回该邮箱对应的用户密码，否则给出用户相应的提示(未找到该用户)。

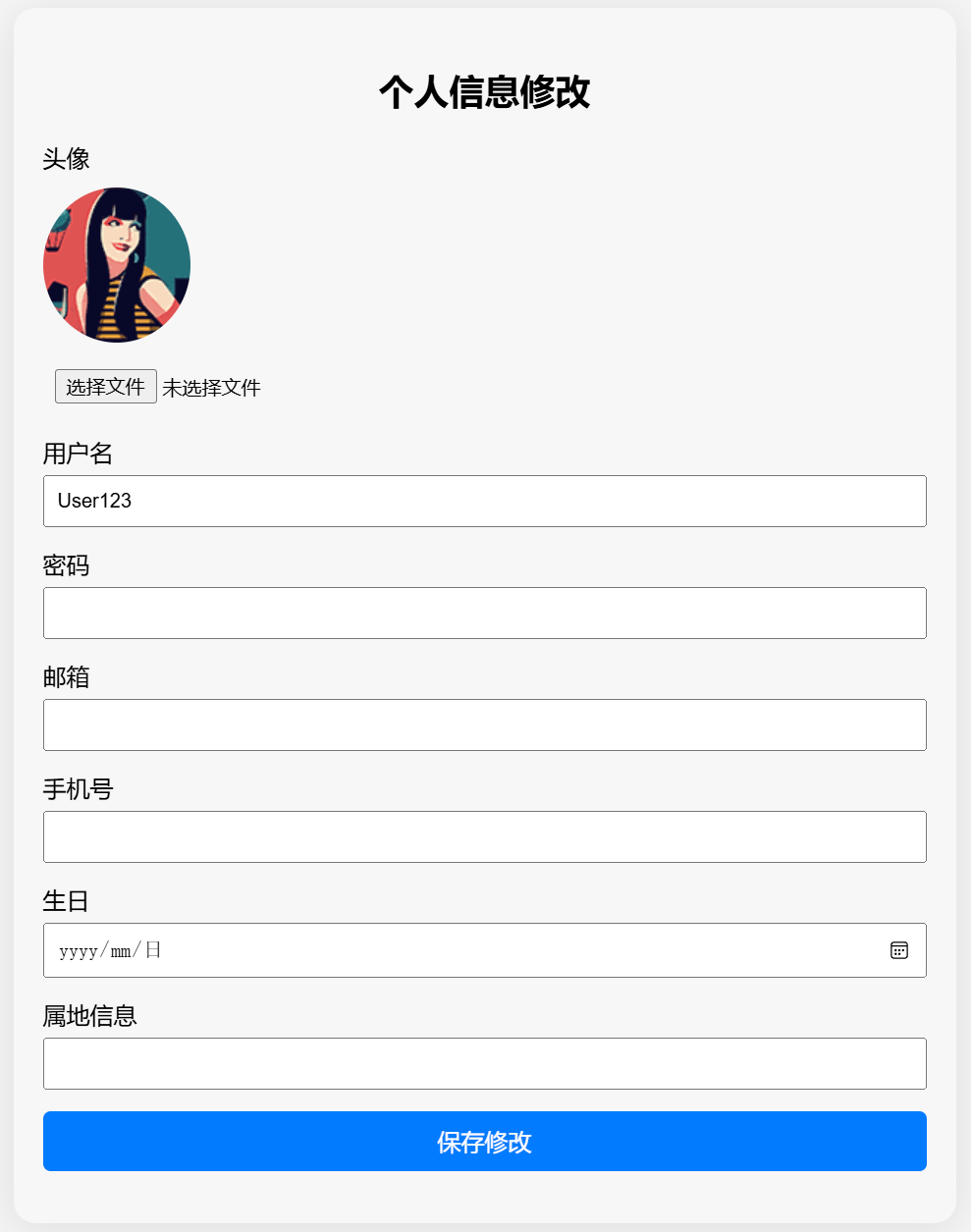
##### 3.用户信息管理

为了方便用户查看其他用户的联系方式等信息，该部分会负责管理每个用户的个人信息。当用户登录进入平台后，用户可以选择进入个人主页，在这里对个人信息进行填写与修改(如手机号，生日，属地等公开信息以及邮箱，密码等私密信息)。信息管理详细情况如表b.4所示

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 输入项 | 具体要求 | 是否必填 |
| 1 | 头像 | 图片从本地上传，文件大小小于2M | 否 |
| 2 | 用户名 |  | 不可填 |
| 3 | 密码 | 由6-12位字符，数字组成 | 否 |
| 4 | 邮箱 |  | 否 |
| 5 | 手机号 |  | 否 |
| 6 | 生日 |  | 否 |
| 7 | 属地信息 |  | 否 |

表b.4 用户信息修改输入项设计说明表

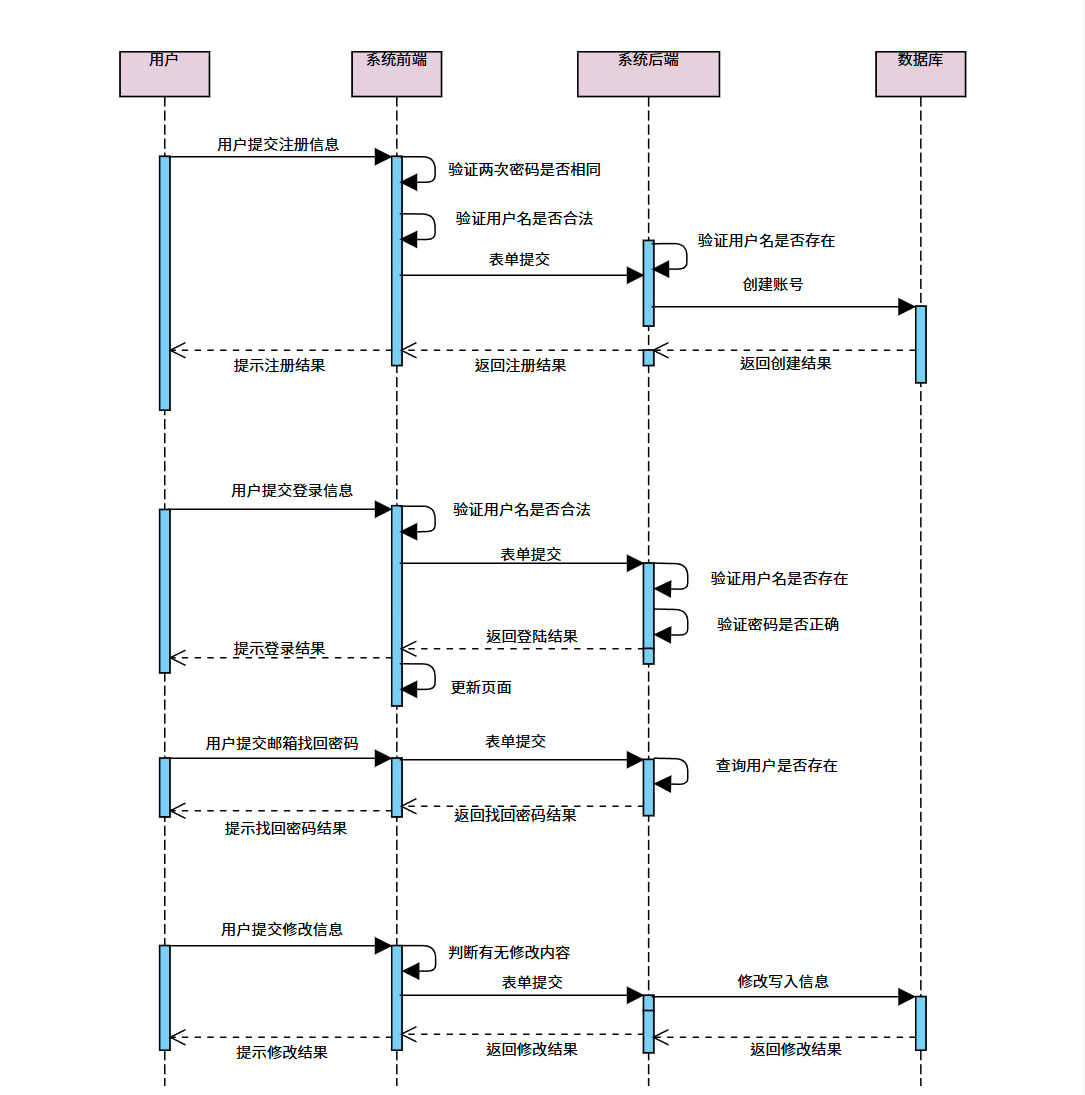
用户信息修改界面UI如图x.6所示



图x.6 用户找回密码界面设计图

系统依照表b.4中的具体要求对用户提交的表单信息进行合法校验，若信息不合法则给出用户提示，指出非法内容。合法则修改用户具体信息。表单中的各项均非必填项，用户可以根据自己的情况选择性的编辑自己的个人信息。

用户管理模块功能的详细UML序列图如图x.7所示



图x.7 用户管理功能模块序列图

#### 2.2.2.2 帖子管理模块设计

在本功能模块中，用户将享有在平台的多个关键区域(包括交友组局、失物招领、以及二手交易)发布个性化需求帖子的权限。用户可以提供详尽的描述信息，并附上有效的联系方式，以便其他用户能够快速响应。此外，用户还有权对已发布的帖子进行实时的编辑、更新，或在必要时删除那些不再需要的帖子，确保信息的时效性和准确性。

为了更直观地展示帖子管理的业务流程，我们提供了帖子管理模块类图（如图x.8所示）。从图中可以清晰地看出，参与此业务功能模块的主要角色包括普通用户和管理员。普通用户负责创建和管理自己的帖子，而管理员则负责对这些帖子进行审核，确保它们符合平台的发布标准。这种分工合作的模式，不仅提高了管理效率，也优化了用户体验。



图x.8 帖子管理类图

从图x.1平台功能模块示意图中可以看到，帖子管理模块主要包括了发布帖子，编辑帖子，修改帖子以及管理员审核帖子四个部分，其中具体每部分功能的详细设计如下。

##### 1.发布帖子

该模块允许用户编辑自己的帖子内容，选择相对应的类型(交友组局，失物招领或二手交易)并发布到平台上。同时，该模块要求用户输入的具体内容有帖子标题，帖子封面(图片)，帖子正文内容以及帖子类别，详细输入项情况如表b.5所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 输入项 | 具体要求 | 是否必填 |
| 1 | 帖子标题 |  | 是 |
| 2 | 帖子封面 | 图片从本地上传，文件大小小于2M | 否 |
| 3 | 正文内容 |  | 是 |
| 4 | 帖子类别 | 提供三种类别供用户选择 | 是 |

表b.5 发布帖子信息输入项设计说明表

发布帖子界面UI如图x.9所示



图x.9 发布帖子界面设计图

系统依照表b.5中的具体要求对用户提交的表单信息进行合法校验，若信息不合法则给出用户提示，指出非法内容。合法则向后端传递发布帖子请求，后端接收到请求后将该帖子发往管理员，并同时返回用户待审核的信息。审核通过后，用户可以在平台上查看到该帖子。

##### 2.编辑帖子

该模块允许用户重新编辑，更新自己所发布的帖子。用户通过打开自己的帖子管理界面，可以查看自己发布的帖子的详细内容，并决定是否修改。修改后的帖子需重新经过管理员审核。该模块要求用户输入的具体内容有帖子标题，帖子封面(图片)，帖子正文内容以及帖子类别，如果用户未填入任何修改内容，则默认不进行任何操作。用户进行修改操作时，若帖子处于"审核中"状态，则用户无法修改该帖子。详细输入项情况如表b.6所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 输入项 | 具体要求 | 是否必填 |
| 1 | 帖子标题 |  | 否 |
| 2 | 帖子封面 | 图片从本地上传，文件大小小于2M | 否 |
| 3 | 正文内容 |  | 否 |
| 4 | 帖子类别 | 提供三种类别供用户选择 | 否 |

表b.6 修改编辑帖子输入项设计说明表

编辑帖子界面UI如图x.10所示



图x.10 详细编辑界面设计图

系统先检查帖子状态，若不处于"审核中",则可执行操作，再依照表b.6中的具体要求对用户提交的表单信息进行合法校验，不合法则向用户返回非法提示(图片过大等),否则向管理员提交审核，经过审核会将修改后的帖子重新发布到平台上。

##### 3.删除帖子

该模块允许用户删除出错或已经不需要的帖子，确保信息的时效性和准确性。当用户在个人帖子界面对个人发布的内容进行管理时，点击帖子的删除按钮，即可删除对应的帖子。为了防止用户误操作，系统设计了二次确认机制，要求用户在删除前明确确认其意图。删除操作完成后，系统会立即向用户显示一个成功或失败的提示信息，确保用户了解操作结果。 同时，用户只能操作已发布的帖子，对处于审核中的帖子，普通用户没有操作权限。

个人帖子界面UI如图x.11所示



图x.11 个人帖子界面设计图

系统对于用户的删除操作，首先会在前端弹窗进行二次确认。确认删除动作后，系统会检查帖子状态，若可删除，则向后端请求删除该帖子数据，否则向用户返回非法操作提示(无权限)。

##### 4.审核帖子

审核帖子是确保平台内容质量和合规性的关键环节。在这个模块中，所有用户发布的帖子在公开展示之前，都必须经过一个严格的审核流程。帖子一旦提交，系统会自动将其发送到后台管理系统，进入待审核队列。

管理员在收到帖子后，将根据一系列预设的标准和规则进行细致的审核。一旦帖子通过审核，它将被正式发布到用户平台上，供所有社区成员查看和互动。这不仅保证了内容的质量和可靠性，也维护了平台的良好秩序和声誉。对于未能通过审核的帖子，管理员会将其状态更新为"未通过"，并根据情况决定是否通知用户具体的拒绝原因，以便用户了解并采取相应的行动。该模块要求管理员输入每个帖子的最终状态("已通过"，"未通过")以及具体原因(选填)。具体输入项设计如表b.7所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 输入项 | 具体要求 | 是否必填 |
| 1 | 最终状态 | 已通过/未通过两种状态可选 | 是 |
| 2 | 具体原因 |  | 否 |

表b.7 审核帖子输入项设计说明表

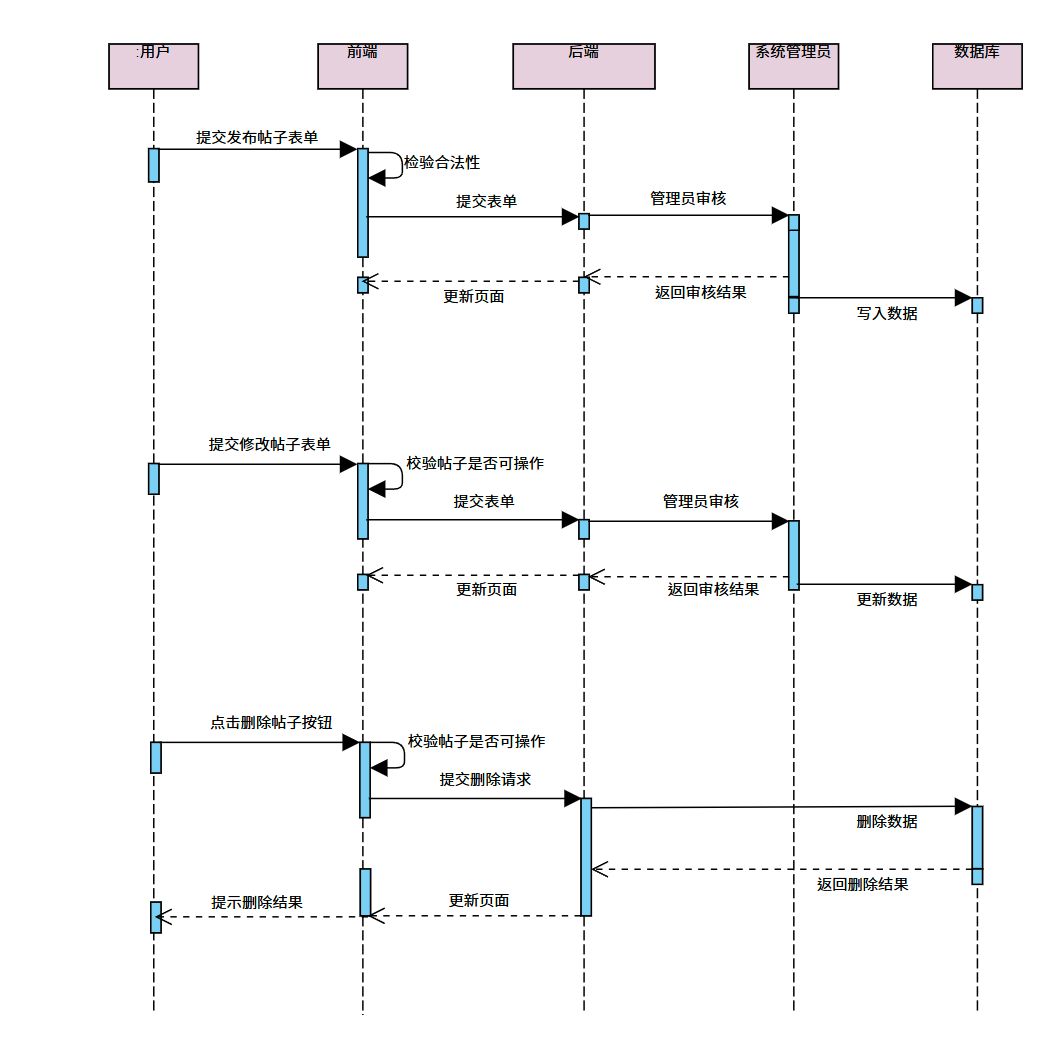
待审核队列界面UI如图x.12所示



图x.12 待审核队列界面设计图

管理员可以通过查看按钮查看帖子详细内容，确认帖子内容符合规范无误后，可以选择将状态选为"已通过"，否则选为”未通过“并可以附上具体原因。当管理员对一个帖子审核完毕后，可以点击提交按钮，代表该帖子条目处理完毕，由系统决定是否将其发布到平台。

帖子管理模块功能的详细UML序列图如图x.13所示

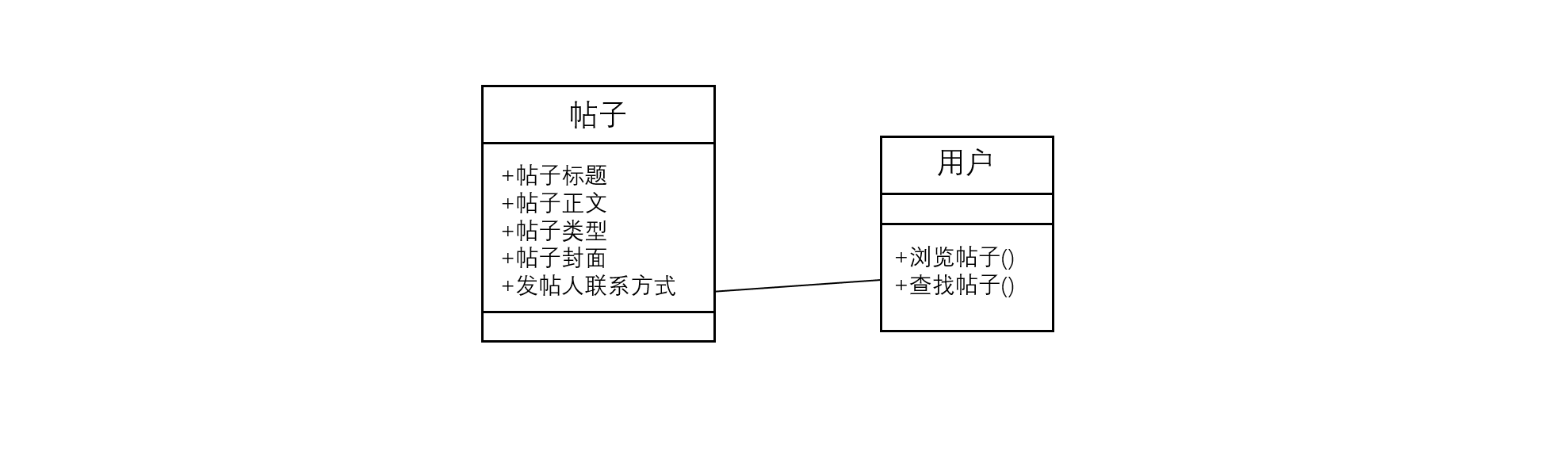


图x.13 帖子管理功能模块序列图

#### 2.2.2.2.3 帖子浏览模块设计

在本功能模块中，用户将享有在平台的多个关键区域(包括交友组局、失物招领、以及二手交易)浏览帖子的权限。用户可以选择自己感兴趣的区域，浏览帖子详细信息，并方便的与发帖人取得联系，进行沟通和交易。此外，用户还可以通过首页导航栏搜索符合自己需求的帖子，提供更友好便捷的使用体验。

此外，为了更直观地展示帖子浏览相关的业务流程，我们提供了详细的帖子浏览模块类图（如图x.14所示）。从图中可以清晰地看出，参与此业务功能模块的主要角色是普通用户，他们通过平台进行信息的浏览、交流和交易。类图详细展示了用户与帖子之间的交互关系，以及用户如何通过不同的功能模块实现自己的需求。



图x.14 帖子浏览类图

从图x.1平台功能模块示意图中可以看到，帖子浏览模块主要包括了浏览帖子内容，查找帖子以及联系发帖人三个功能，具体每部分功能的详细设计如下。

##### 1.浏览帖子内容

该功能模块允许用户根据自己的兴趣和需求，选择关注特定的区域或帖子类型，实现个性化的内容浏览。同时，每个帖子都提供了详尽的信息，包括活动或商品的描述、图片、价格、时间、地点等，帮助用户全面了解帖子内容。当用户点击帖子时，便可以进入帖子详细页，浏览有关内容。

帖子详细页面UI如图x.15所示



图x.15 帖子详细界面设计图

用户可以在此页面查看详细的帖子内容，根据发帖人提供的联系方式与其获得联系，也可点击返回主页按钮浏览其他感兴趣的帖子内容。

##### 2.查找帖子

该功能模块允许用户根据自己的特定需求，在交友组局、失物招领以及二手交易这三个关键模块中做出选择，并进行基于关键词的模糊查找。这种设计旨在简化用户的搜索过程，同时确保他们能够快速定位到自己感兴趣的帖子。

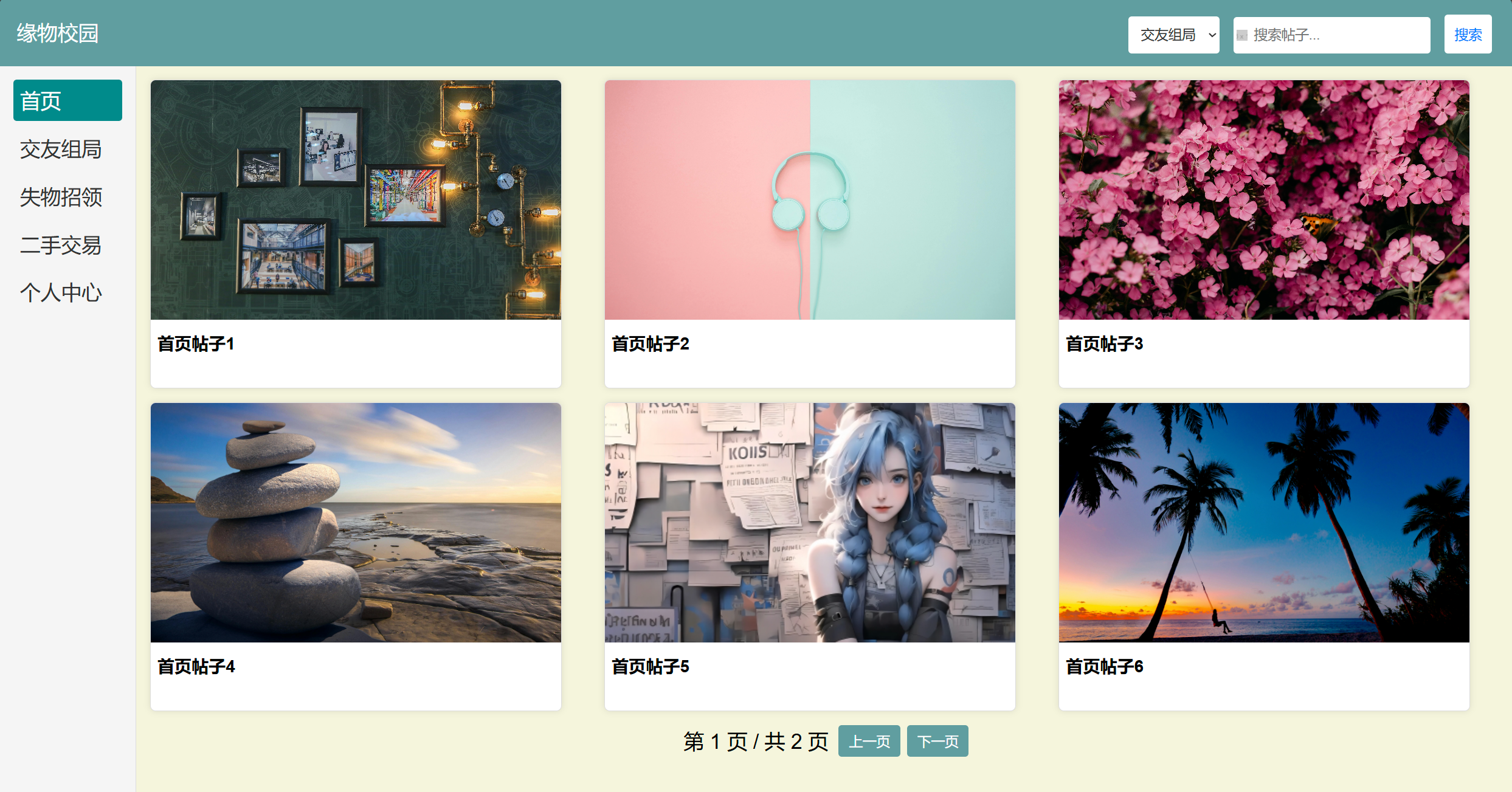
用户在使用查找功能时，首先会看到一个下拉列表，分别对应平台的三个主要模块。他们可以根据自己的目的，选择进入交友组局寻找社交活动，或是进入失物招领寻找丢失的物品，亦或是进入二手交易寻找心仪的商品。一旦选定模块，用户就可以输入关键词进行搜索，系统将根据这些关键词，展示所有相关的帖子。

我们的模糊查找技术确保了搜索结果的相关性，即使用户输入的关键词与帖子内容不完全匹配，系统也能够智能识别并推荐最接近的帖子。这意味着用户无需担心关键词的精确性，只需输入大致的描述，即可获得一系列可能的选项，为用户提供了友好便捷的服务。查找帖子页面输入项具体设计如表b.8所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 输入项 | 具体要求 | 是否必填 |
| 1 | 帖子类别 | 在交友组局/失物招领/二手交易的下拉框里选择 | 是 |
| 2 | 关键词 |  | 否 |

表b.8 查找帖子输入项设计说明表

查找帖子界面UI如图x.16所示



图x.16 查找帖子界面设计图

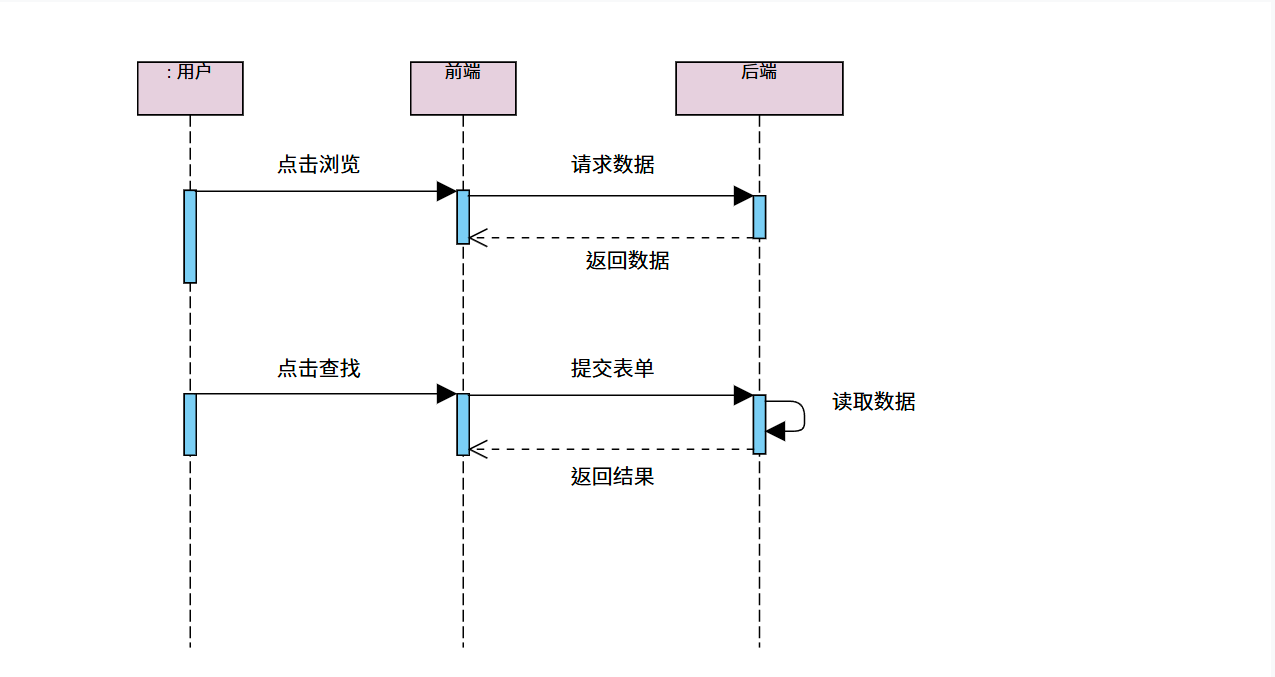
在界面右上角，用户可以选择想搜索的帖子类别，并在输入框内输入关键字进行搜索。当点击搜索按钮后，系统会进行模糊查询，将一系列符合关键词的对应帖子呈现在用户当前页面中。

##### 3.联系发帖人

由图x.15帖子详细页面设计图，我们可以看到发帖人会将联系方式明确注明在帖子正文中。用户可以直接通过该联系方式与发帖人取得联系，无需经过任何中间环节。本平台致力于提供一个透明、直接的用户交流环境，确保双方的隐私安全和沟通效率。

为了进一步促进用户间的交流，我们还计划引入一些社交功能，比如用户可以根据帖子内容发起讨论，或者对发帖人的其他帖子进行浏览和评论。这样的设计不仅能够增加用户粘性，还能够激发社区内的互动和讨论，形成良好的社区氛围。

帖子浏览模块功能的详细UML序列图如图x.17所示



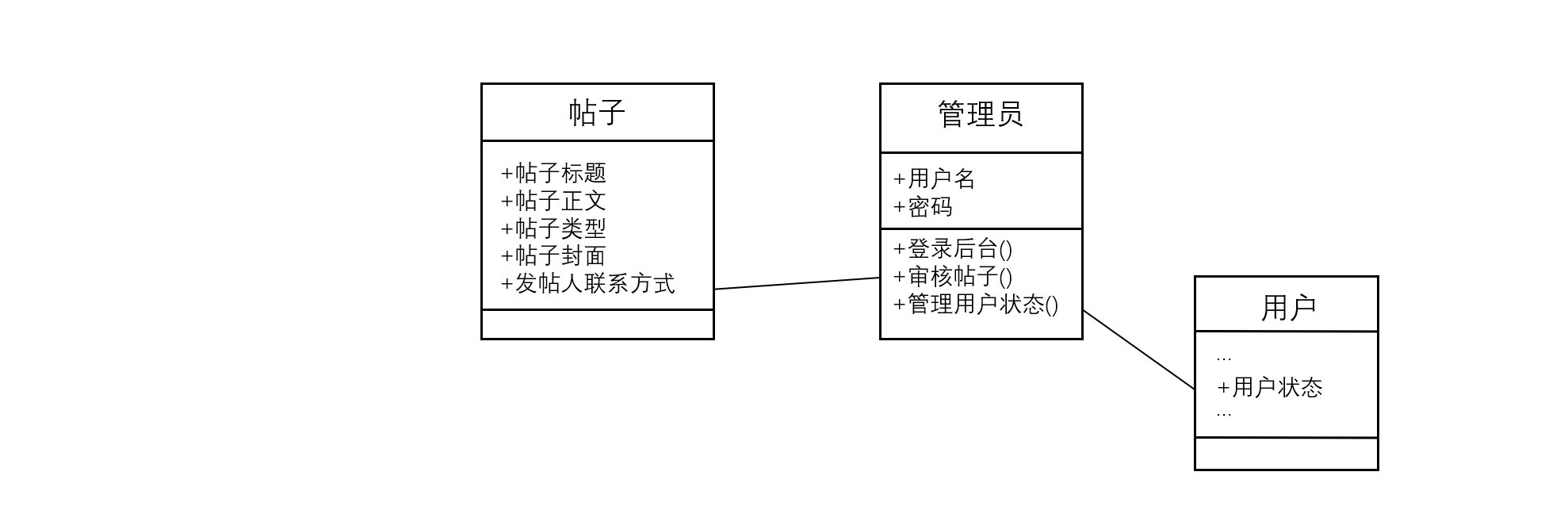
图x.17 帖子浏览功能模块序列图

#### 2.2.2.2.4 管理员管理模块设计

管理员管理模块设计是确保平台健康、有序运行的关键环节。在本功能模块中，管理员可以使用管理员账号登录系统的后台管理界面，拥有对整个平台内容和用户行为的监控与调控能力。

为了更直观地展示管理员管理的业务流程，我们提供了管理员管理模块类图（如图x.18所示）。从图中可以清晰地看出，管理员在系统中扮演着监管者和协调者的角色，他们的操作直接影响着平台的运行质量和用户体验。类图详细展示了管理员与用户、帖子以及其他系统组件之间的交互关系，包括权限验证、帖子审核流程、用户账号状态管理等关键环节。

通过这个模块，管理员可以高效地完成日常管理工作，及时发现并解决潜在问题，保障平台的稳定运行和用户的合法权益。同时，这也为平台的持续改进和优化提供了有力支持。



图x.18 管理员管理类图

从图x.1平台功能模块示意图中可以看到，管理员管理模块主要包括了管理员登陆，帖子审核管理以及用户状态管理三个部分，其中具体每部分功能的详细设计如下。

##### 1.管理员登录

管理员登录是进入管理模块的第一步。在这一环节，系统提供了一个安全、高效的登录界面，要求管理员输入用户名和密码进行身份验证。为了区分普通用户和管理员，会为所有账号拥有者提供一个权限字段，该字段为1表示为管理员。登录成功后，管理员将进入后台管理界面，获得对后台管理系统的访问权限，从而能够执行各种管理任务。登录信息及详细情况如表b.9所示

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 输入项 | 具体要求 | 是否必填 |
| 1 | 用户名 | 只能由6-16位字母，数字组成 | 是 |
| 2 | 密码 |  | 是 |

表b.9 管理员登录信息输入项设计表

管理员登录界面UI如图x.19所示(与普通用户登录界面一致)



图x.19 用户登录界面设计图

前端使用axios对管理员提交的表单进行拦截与校验，如果管理员输入的登录信息不合法(用户名格式错误)，则给出提示，告知其非法原因。若合法则交由后端处理，对信息进行检验。登录成功后主动跳转到后台管理页面。

##### 2.帖子审核管理

帖子审核管理是管理员职责中的核心部分。管理员需要对用户提交的帖子内容进行仔细审查，确保它们不违反平台的社区准则和法律法规。并给予管理员相应的提示。管理员可以批准或拒绝发布，还可以设置关键词过滤和敏感词库，以提高审核效率。此外，管理员还可以查看帖子的审核历史和统计数据，以监控审核工作的质量和效率。

其中，具体的审核功能实现以及界面UI设计可见[帖子管理模块-4.审核帖子](#jump)

##### 3.用户状态管理

用户状态管理是管理员维护平台秩序的重要工具。管理员可以查看用户的注册信息、对用户账号进行全面的监控和管理。在这一环节，当管理员检测到用户行为异常或违反规定时，管理员可以采取相应的措施，包括但不限于暂时冻结或永久封禁账号。

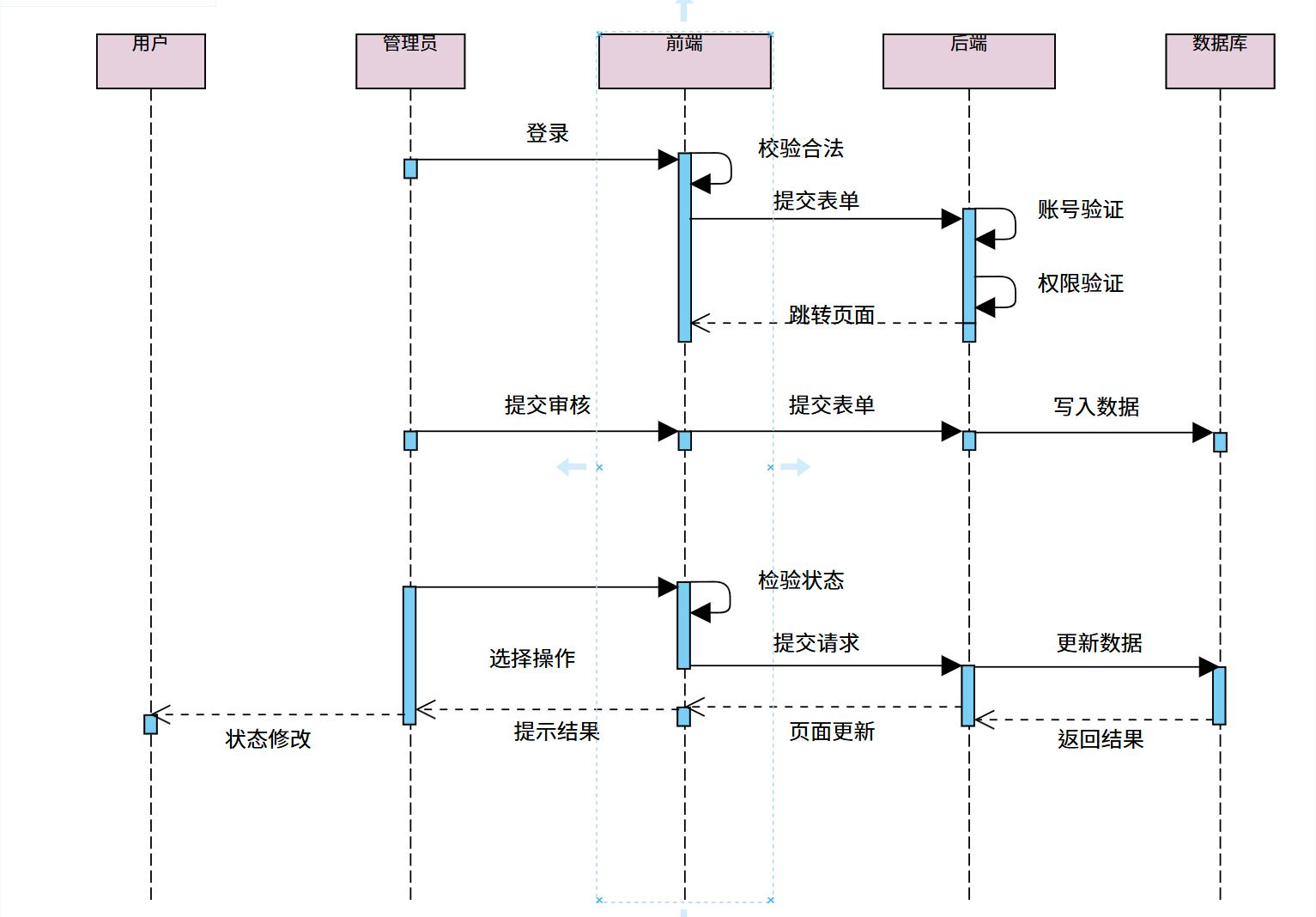
用户状态管理界面UI如图x.20所示



图x.20 用户状态管理界面设计图

管理员进行用户状态操作，首先会在前端弹窗进行二次确认。确认操作后，系统会检查用户状态，若用户不处于当前操作所指向的状态，则可变更。否则拒绝变更，告知管理员信息(用户已处于该状态)。

管理员管理模块功能的详细UML序列图如图x.21所示



图x.21 管理员管理功能模块序列图

## 2.3 数据库设计

数据库设计是确保平台数据信息存储和用户帖子关系管理有效性的关键步骤。在本小节中，我们将根据系统功能模块的设计，构建一个结构合理、高效且可扩展的数据库，以支持系统业务的多个不同功能模块。

系统数据库的设计分为逻辑结构设计和物理结构设计两个方面。逻辑结构设计是数据库设计过程中的首要步骤。它涉及定义数据的逻辑关系，包括确定实体、实体之间的关系以及数据的属性。此外，我们还需要考虑表之间的关系，如一对多或多对多关系，并通过外键来实现这些关系。物理结构设计则关注数据库在存储介质上的实际布局。这包括确定索引策略来优化查询性能和选择合适的存储引擎以满足不同的读写需求。

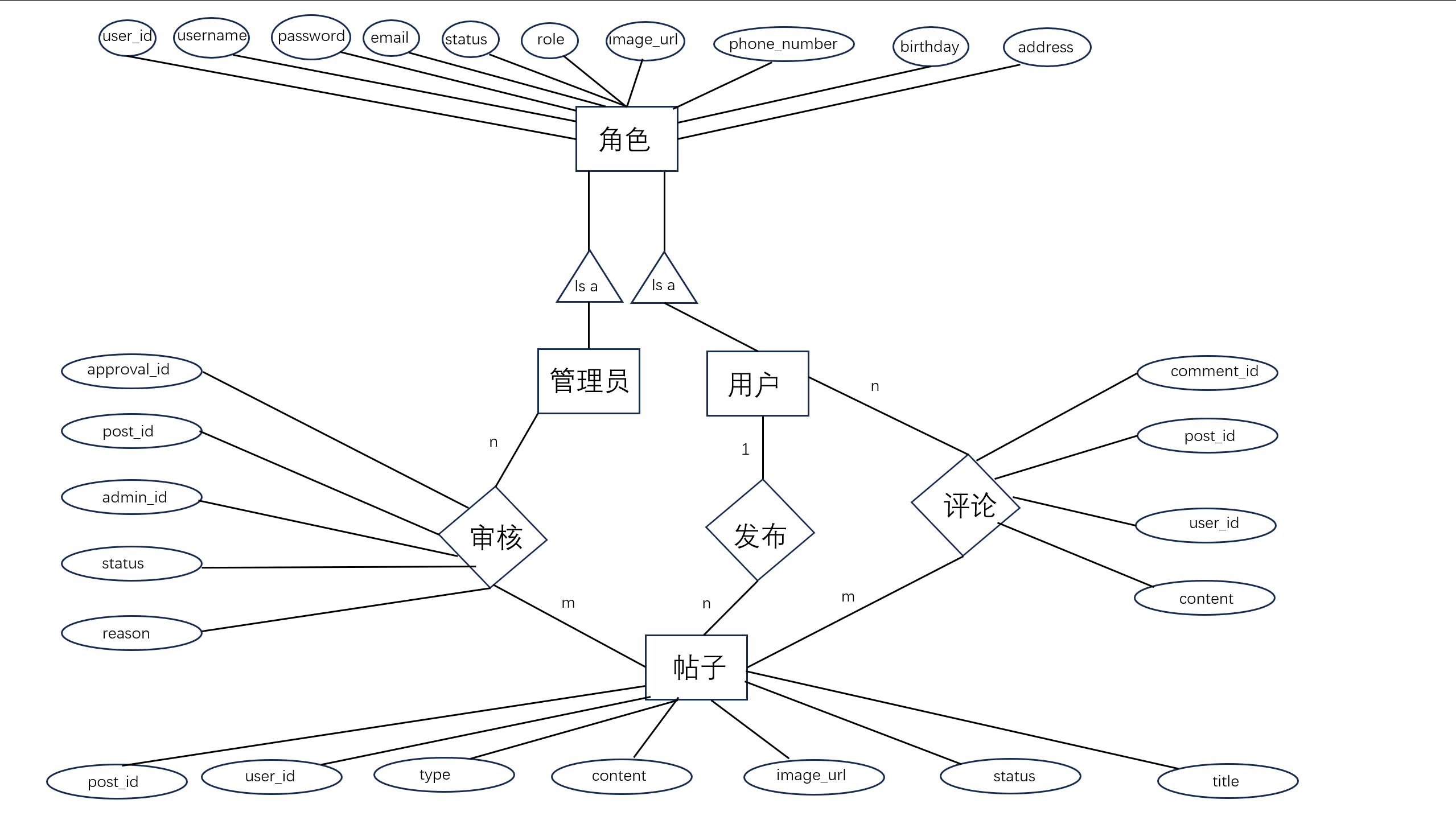
本系统采用MySQL作为系统开发数据库。

### 2.3.1 逻辑结构设计

在校园生活服务平台中，数据库的设计对于整个系统的功能实现至关重要。该平台允许用户在不同的板块发布帖子，表达他们的诉求和需要，而管理员则负责审核这些帖子，确保内容的合规性。为了保障数据库的高效运作和数据的准确性，我们首先对逻辑结构进行了精心设计，确立了系统中各个数据库实体的关系。

针对校园生活服务平台，我们设计的数据库实体涵盖了用户、帖子、板块、管理员等核心组件。每个实体都有其特定的属性信息，以及与其他实体之间的逻辑联系。

通过具体的E-R图（实体-关系图），我们能够清晰地展示出这些数据库实体的属性信息以及它们之间的联系。本平台详细E-R图如图x.22所示



图x.22 校园生活服务平台E-R图

### 2.3.2 物理结构设计

根据上文对系统数据库的逻辑设计以及数据库设计基本理论，校园生活服务平台主要数据用表如表b.10所示。

|  |  |  |
| --- | --- | --- |
| 序号 | 数据表 | 数据表描述 |
| 1 | users | 用户信息表 |
| 2 | posts | 帖子信息表 |
| 3 | posts\_approvals | 审核信息表 |
| 4 | comments | 评论信息表 |

表b.10 系统主要数据表清单

系统关键数据表设计详情如下：

1.用户信息表，表名为users，数据设计详情如表b.11所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 字段类型 | 可空 | 默认值 | 主键 |
| user\_id | 用户id | int | 否 | 无 | 是 |
| username | 用户名 | varchar(50) | 否 | 无 | 否 |
| password | 密码 | varchar(255) | 否 | 无 | 否 |
| email | 邮箱 | varchar(255) | 否 | 无 | 否 |
| status | 帐号状态 | enum('normal','frozen','banned') | 否 | ’normal‘ | 否 |
| role | 角色 | enum('user','admin') | 否 | ’user‘ | 否 |
| image\_url | 头像URL | varchar(255) | 是 | 无 | 否 |
| phone\_number | 手机号 | varchar(20) | 是 | 无 | 否 |
| birthday | 生日 | date | 是 | 无 | 否 |
| address | 地址 | text | 是 | 无 | 否 |

表b.11 用户信息数据表

2.帖子信息表，表名为posts，数据设计详情如表b.12所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 字段类型 | 可空 | 默认值 | 主键 |
| post\_id | 帖子id | int | 否 | 无 | 是 |
| usee\_id | 用户id | int | 否 | 无 | 否 |
| type | 帖子类型 | enum('friends','lost','trade') | 否 | 无 | 否 |
| title | 帖子标题 | varchar(255) | 否 | 无 | 否 |
| content | 帖子内容 | text | 否 | 无 | 否 |
| image\_url | 帖子封面 | varchar(255) | 是 | 无 | 否 |
| status | 帖子状态 | enum('pending','approved','rejected') | 是 | ’pending‘ | 否 |

表b.12 帖子信息数据表

3.审核信息表，表名为posts\_approvals，数据设计详情如表b.13所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 字段类型 | 可空 | 默认值 | 主键 |
| approval\_id | 审核id | int | 否 | 无 | 是 |
| post\_id | 帖子id | int | 否 | 无 | 否 |
| admin\_id | 管理员id | int | 否 | 无 | 否 |
| status | 帖子状态 | enum('approved','rejected') | 否 | 无 | 否 |
| reason | 审核原因 | text | 是 | 无 | 否 |

表b.13 审核信息数据表

4.评论信息表，表名为comments，数据设计详情如表b.14所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 字段类型 | 可空 | 默认值 | 主键 |
| comment\_id | 评论id | int | 否 | 无 | 是 |
| post\_id | 帖子id | int | 否 | 无 | 否 |
| user\_id | 用户(评论者)id | int | 否 | 无 | 否 |
| content | 评论内容 | text | 否 | 无 | 否 |

表b.14 评论信息数据表