

校园公告发布平台“青言速递”详细设计文档

版本： V1.0

名字： 青言速递

日期： 2025-10-18

编写人： 胡博潇

1. 文档概述

1.1 编写目的

本详细设计文档基于《软件需求规格说明书》和《架构设计草案》，对“青言速递”系统的各功能模块进行深入的设计说明。

其目的是为开发人员提供具体的编码依据，为测试团队提供功能与逻辑参考，并确保项目组成员对系统实现逻辑与模块交互有统一的理解。

1.2 设计范围

本文档的设计范围涵盖系统后端的核心业务逻辑模块，主要包括：

- 用户管理模块
- 公告管理模块
- 分类与检索模块
- 评论与统计模块

前端交互逻辑、样式规范及CI/CD流水线脚本不在本文档的描述范围之内。

2. 系统概述

“青言速递”旨在构建一个统一、高效的校园信息发布与传播平台，为师生提供可靠的公告发布与获取渠道。

系统采用前后端分离的 **B/S 架构**，主要技术选型如下：

- 后端框架：** Spring Boot
- 前端框架：** Vue 3
- 数据库：** MySQL
- 缓存：** Redis
- 部署方式：** Docker 容器化

系统设计重点关注可维护性、响应速度与数据安全性。

3. 架构与设计决策回顾

3.1 技术架构决策

决策点	方案选择	原因说明
应用架构	Spring Boot 单体应用	初期系统规模较小，便于快速开发与维护
鉴权方式	JWT + Spring Security	无状态认证，提升并发性能与扩展性
缓存机制	Redis 缓存公告列表与热点数据	降低数据库查询压力，加快首页响应速度
部署方式	Docker 容器化	保证部署环境一致性，支持快速扩展

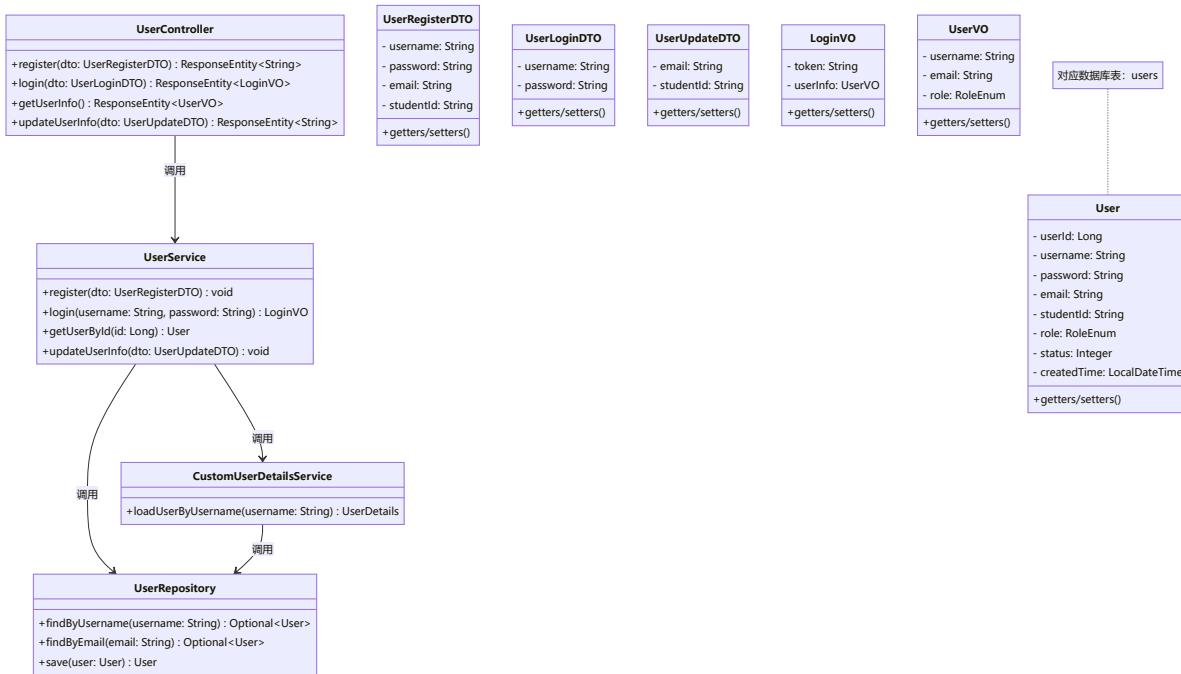
4. 模块详细设计

4.1 用户管理模块 (FU-01)

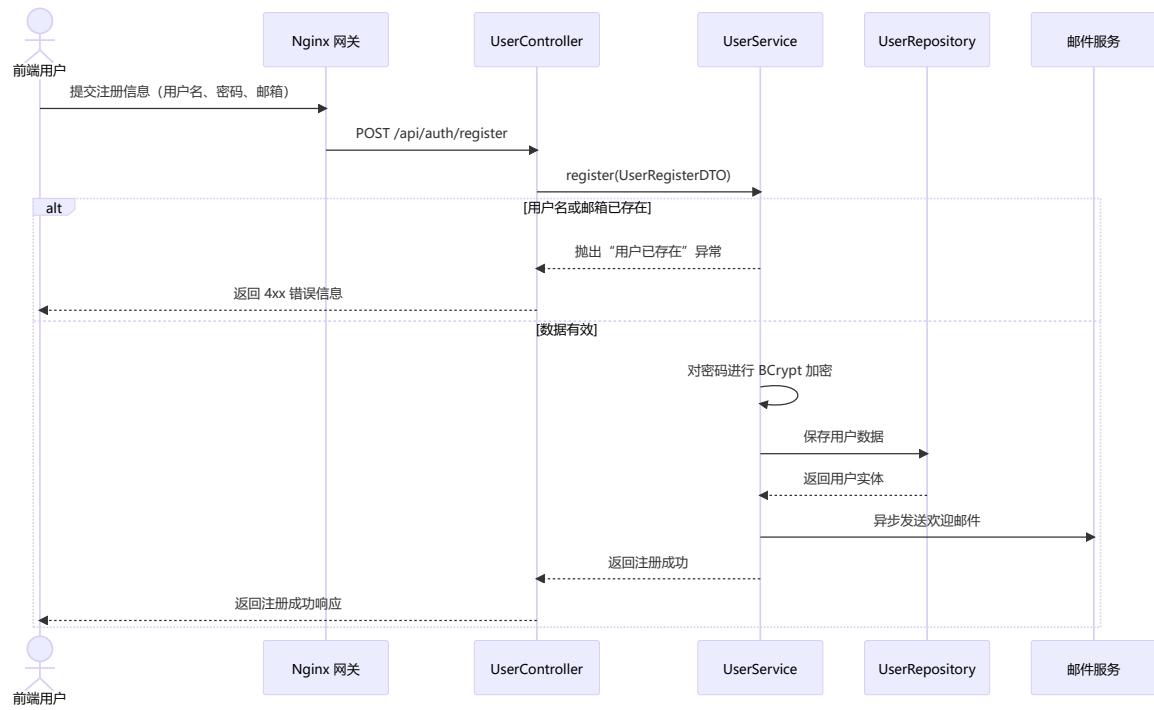
4.1.1 模块描述

用户管理模块负责系统的注册、登录、个人信息维护及角色权限管理。后台管理员可通过该模块进行用户信息的查询与状态管理。

4.1.2 类图



4.1.3 核心流程设计 - 用户注册

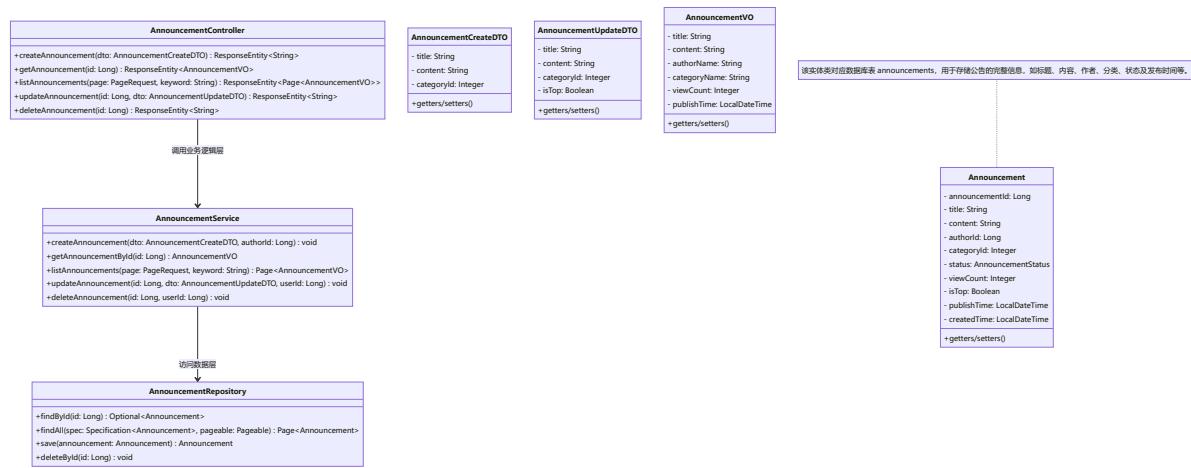


4.2 公告管理模块 (FU-02)

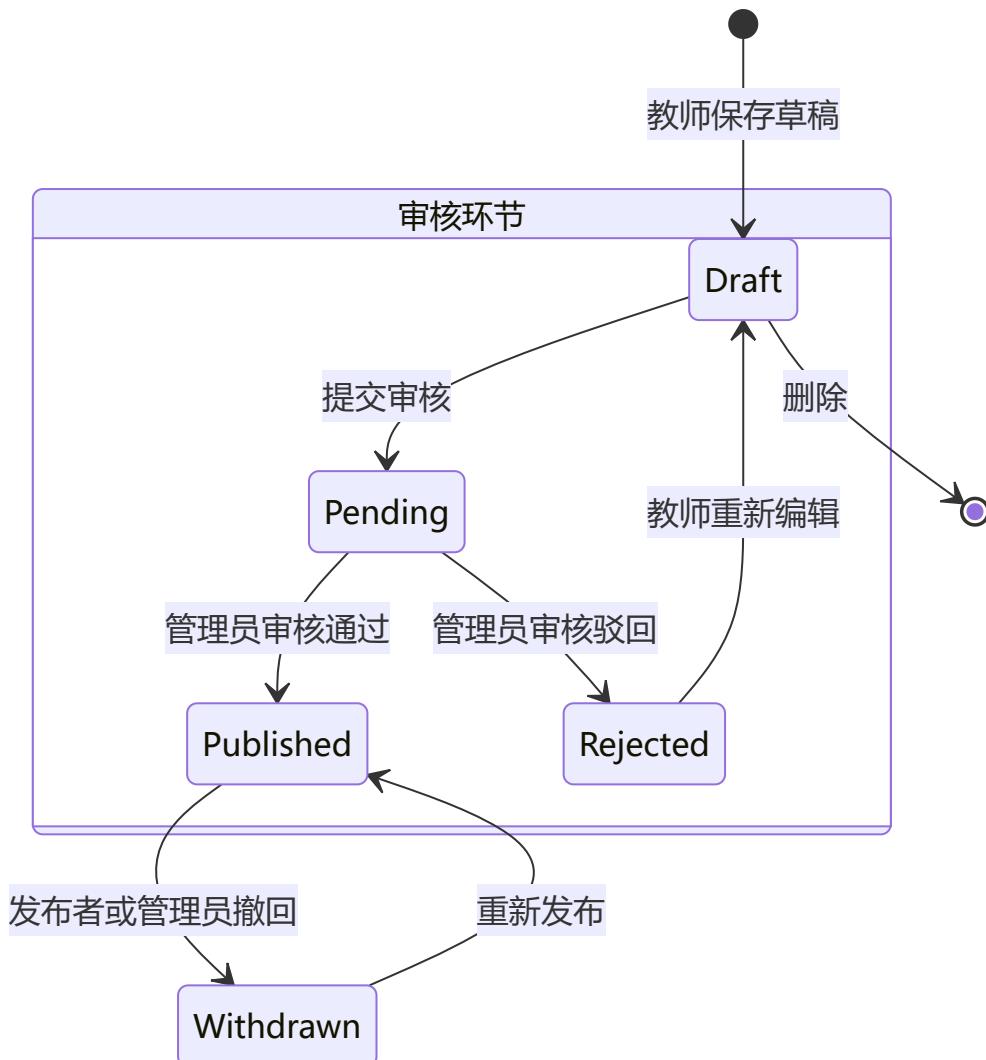
4.2.1 模块描述

处理公告的创建、编辑、删除、审核、发布与列表展示等全生命周期管理。

4.2.2 类图



4.2.3 核心流程设计 - 公告创建与审核



4.3 分类与检索模块 (FU-03)

4.3.1 模块描述

负责公告分类的维护，以及公告的列表分页展示和关键词搜索。

4.3.2 核心接口设计

- 公告列表与搜索接口
 - 路径: GET /api/announcements
 - 参数:
 - page: 页码 (默认0)
 - size: 每页大小 (默认10)
 - keyword: 搜索关键词 (可选)
 - categoryId: 分类ID (可选)
 - 逻辑:
 1. 检查是否存在keyword和categoryId。
 2. 若参数为空，尝试从Redis读取缓存的首页公告列表。
 3. 若缓存未命中，则构建JPA查询条件，查询数据库。
 4. 将结果转换为VO并返回，首页结果同时写入Redis缓存。

4.4 评论与统计模块 (FU-05)

4.4.1 模块描述

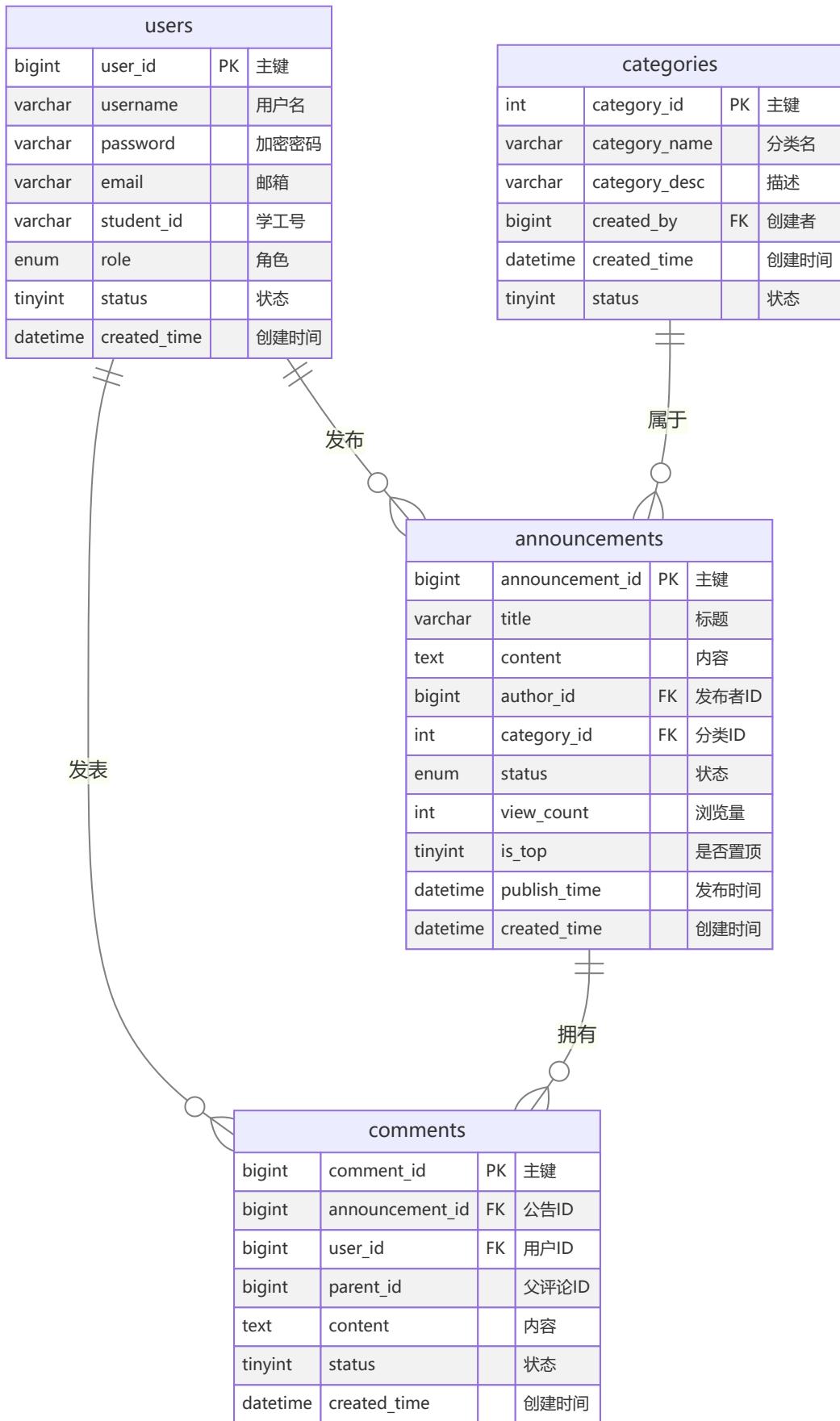
处理公告评论的增删改查，以及公告浏览量的统计。

4.4.2 浏览量统计设计

- 实现策略：采用Redis异步持久化方案。
 1. 前端请求公告详情时，调用 Redis.INCR(announcement_view_count:{id})。
 2. 将 {公告ID, 最新浏览量} 存入一个Redis List (队列) 中。
 3. 通过定时任务（如每5分钟），从队列中取出数据，批量更新到MySQL的announcements表。

5. 数据库详细设计

5.1 实体关系图 (ERD)



5.2 索引设计

- announcements表：
 - idx_author_status (author_id, status): 加速“我的公告”查询。
 - idx_category_publishtime (category_id, publish_time DESC): 加速按分类和时间排序的列表查询。
 - idx_title_content (全文索引): 支持对标题和内容的全文搜索（若MySQL版本支持）。
- comments表：
 - idx_announcement_time (announcement_id, created_time): 加速加载某公告下的评论。

6. 接口详细设计

6.1 用户登录接口

- 接口路径：POST /api/auth/login
- 请求体：

```
1 | {
2 |     "username": "string",
3 |     "password": "string"
4 | }
```

- 成功响应 (200 OK)：

```
1 | {
2 |     "code": 200,
3 |     "message": "success",
4 |     "data": {
5 |         "token": "eyJhbGciOiJIUzI1NiI6IkpXVCJ9...",
6 |         "userInfo": {
7 |             "userId": 1,
8 |             "username": "张三",
9 |             "role": "teacher"
10 |         }
11 |     }
12 | }
```

- 失败响应 (401 Unauthorized)：

```
1 | {
2 |     "code": 401,
3 |     "message": "用户名或密码错误",
4 |     "data": null
5 | }
```

6.2 创建公告接口

- 接口路径: POST /api/announcements
- 请求头: Authorization: Bearer {token}
- 请求体:

```
1  {
2      "title": "新学期开学通知",
3      "content": "...",
4      "categoryId": 1
5 }
```

- 成功响应 (200 OK):

```
1  {
2      "code": 200,
3      "message": "公告创建成功，等待审核",
4      "data": null
5 }
```

7. 非功能性需求设计

- 性能:
 - 公告列表接口使用Redis缓存第一页数据，缓存失效时间为5分钟。
 - 浏览量统计采用Redis INCR命令，每累积100次浏览量再异步写回MySQL。
- 安全:
 - 所有/api/admin

路径的请求需校验用户角色为ROLE_ADMIN。

- 公告的编辑、删除接口需校验当前用户是否为公告作者或管理员。
- 所有用户输入（公告内容、评论）进行XSS过滤。

8. 总结

本详细设计文档在架构草案的基础上，细化并固化了核心模块的实现方案。通过UML图（类图、序列图、状态图、ER图）和接口定义，为后续开发提供了明确的“施工蓝图”。