## 前言

本文是学生管理系统详细架构设计文档，用于指导学生管理系统后续的开发、测试和运维。

## 修订历史

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 主要内容 | 修订者 | 日期 |
| V0 | 创建文档，内容编写 | 梁世雄 | 2022.10.26 |

## 词汇表

Reactor: 网络编程模式

Netty: 开源的网络编程框架

## 1. 业务背景

为了开发一个学生管理系统。为便于模块功能划分以及后期维护，可以拆分成三个子系统，学生管理子系统、课程子系统和权限子系统。这三个子系统访问量不会很频繁，人数一般不超过万人，每年的人数相对稳定，所以不存在高性能高并发的情况。所以对性能要求并不高。普通数据库就能满足要求，考虑可用性的问题，数据库采用主备。

## 2. 约束和限制

1. 人数不超过 1 万人
2. 成本控制在 50 万以内
3. 使用 MySQL 数据库，并使用主备库
4. 拆分成三个子系统
5. 三个月内完成

## 3. 总体架构

### 3.1 架构分析

复杂度分析

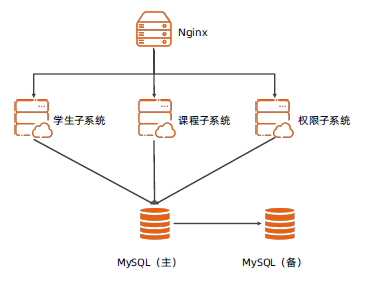
由于学生管理系统面向的是学校，操作上不会很频繁和同时大并发的访问，所以性能这块不作为考核点。而且学生信息只要不是大量丢失，也都是可以理解的，手工再录入也不是可以接受的。也不涉及到复杂的逻辑关系，所以整体没什么难度。

### 3.2 总体架构

只有学生，老师和课程三者之间的关系，相对比较简单。一个学生可以选择多个老师的课程，一个老师也可以被多个学生选择，所以是多对多的关系。

这里使用 MySQL 就能满足要求，一主一备，主备数据同步采用同步或异步方式均可。

学生子系统管理学生的相关信息，姓名，学号，性别，年龄，班级之类的数据。课程子系统大概包含上课时间，课程，老师，学生等信息。权限管理子系统包括学生，老师个人信息的修改以及课程的修改。



## 4. 详细设计

### 4.1 核心功能

4.1.1 学生子系统

学生子系统用于显示学生的个人信息，老师的信息也划归到学生子系统，完成个人信息录入以及修改。

4.1.2 课程子系统

课程子系统是老师和学生的纽带，课程根据老师的时间排班，老师可以在不同的时间范围授课，学生选择自己喜欢的课程和时间。

4.1.3 权限子系统

权限管理子系统用于设置权限，比如大一的学生只能选大一的课程和对应老师，。

4.1.4 数据库

MySQL 主备方式，用于防止数据丢失。

### 4.2 关键设计

高性能和高可用以及安全都不是重点考虑对象，着重考虑下可扩展吧。拆分为三个子系统就是为了功能划分清晰以及后期便于维护的。

### 4.3 设计规范

*MySQL 使用 Innodb 存储引擎*

*连接池使用* hikariCP

开发语言使用 JAVA 1.8

使用前后端分离

## 5. 质量设计

成本主要在硬件上，三个子系统使用三台服务器或者为了节省成本可以使用一台服务器提供三个服务。数据库主备需要两台机器。还有一台用于 nginx 分发。

其他方面没太多考虑。

## 6. 演进规划

对性能稍微有要求可以从以下几个方面着手。

6.1 因为数据量不大，可以考虑使用内存数据库。

6.2 可以考虑使用线程池以增加子系统的并发访问

6.3 访问数据库会有延时，可以考虑使用 netty 等异步访问方式以增加访问并发量