# 外包学生管理系统架构设计文档

V0.0.1

2022-06-08

## 修订履历

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 变更内容 | 修订者 | 日期 | 评审者 |
| V0.0.1 | 创建 | 某某 | 2022-06-08 | 某某 |

## 前言

本文是外包学生管理系统架构设计文档，用于指导外包学生管理系统的开发、测试及运维。

## 词汇表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 术语 | 缩写 | 定义 |
|  |  |  |  |

## 参考资料

无

## 业务背景

随着学校规模的不断扩大，学生数量的增加，需要处理的信息也日趋增大。学生管理面临一些问题：（1）管理学生信息需要花费大量的老师资源；（2）原始的处理信息效率十分低下。

为提高学生管理的管理水平，优化资源，尽可能降低管理成本成为学生管理的新课题，学生管理系统是从学生管理现状出发，根据学生管理的新要求进行开发设计的，它需要解决学生信息管理数据信息量大修改不方便，对一系列数据进行分析时花费时间长等问题，帮助学生管理人员有效管理学生信息。

因此学生信息管理系统可以通过系统规范地管理、科学性统计和快速查询、修改、增加、删除等，提高信息的准确度以及日常管理的工作效率。

本系统主要是应用于学生各类信息的管理，总体任务是实现学生信息关系的系统化、规范化、自动化、其主要任务是统计学生各类信息进行日常管理、如查询、修改、增加、删除、以及学生选课、成绩的查询等功能设计的管理系统。

## 约束和限制

1、必须在 2022.XX.XX 号完成

2、成本不能超过 10 万

3、数据库采用 MySQL

4、后端使用 JAVA 语言

5、前端使用PHP语言

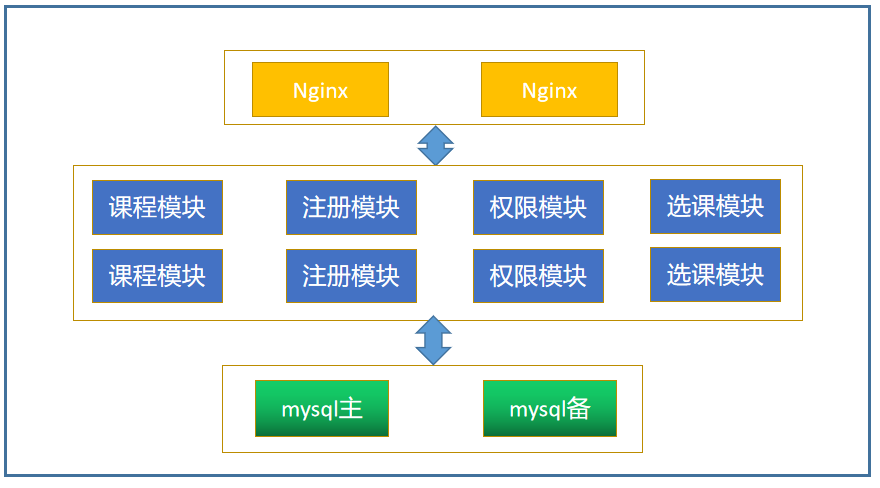
6、系统必须要无状态能支持横向扩展

7、使用Nginx实现负载均衡

8、使用VIP实现Nginx、业务模块的主备切换

9、使用HTTPS

## 3. 总体架构



### 3.1 架构分析

#### 3.1.1高可用

对于学生管理系统，不能让数据丢失，所以在存储层做一个主备近实时同步，保证数据有两份，一份数据丢失时，还有另一份可用的数据，同时在主宕机时，拉起从数据库，尽可能快速恢复，对外提供服务。

每个模块都是双机多活集群，可容忍集群内单点故障。

使用VIP对模块健康状态进行监控，实现负载均衡和故障切换。

#### 3.1.2 扩展性

系统设计成无状态的，很容易横向扩展，以满足不同的性能指标及可用性要求。

可很容易扩展为多机房部署，提升容灾能力。使用mysql备点向备机房同步即可。

扩展新业务，可设计成新模块，对现有系统几乎没影响。

#### 3.1.3 伸缩性

横向，可通过增加、减少服务器及模块数量来提高、降低系统的处理能力。

纵向，可通过增加、减少硬件配置来提高、降低系统的处理能力。

#### 3.1.4 可观测性

监控系统对各模块进程状态进行监控，若有进程退出，及时报警。

监控系统对各服务器状态进行监控，若有服务器宕机，及时报警。

监控系统对各模块异常日志进行实时抓取，若有ERROR、WARN日志，及时报警。

监控系统对各模块事件日志进行实时抓取，WEB端展示系统的实时状态。

#### 3.1.5 安全性

使用HTTPS进行通信，可防止被外网篡改数据包攻击。

## ****4. 详细设计****

### ****4.1****核心功能

#### 4.1.1处理请求流程

客户端请求从客户浏览器发起HTTPS请求，到达系统内部NGINX进行负载均衡，NGINX根据请求类型，转发给业务集群模块；业务模块对此需求进行处理，生成SQL语句访问mysql数据库，并将结果返回给客户端。

### ****4.2关键设计****

#### 4.2.1系统可扩展

根据业务特点将外包学生管理系统拆分成：课程模块、注册模块、权限模块、选课模块。

#### 4.2.2系统高可用

各模块设计成无状态的，将有状态的数据转移给 MySQL 中，没有状态，可以很方便地进行横向扩展，避免系统单点故障，同时提高系统的吞吐。

使用VIP对模块健康状态进行监控，实现负载均衡和故障切换。

使用两台服务器，每台服务器分别部署一套业务模块，即：一个课程模块、一个注册模块、一个权限模块、一个选课模块。

#### 4.2.3存储高可用

Mysql使用主备双点模式，部署在两台服务器，一台用于主数据库，一台用于备数据库，以避免单台服务器宕机。主从考虑异步复制，提升系统的可用性及响应时间。

#### 4.2.4系统高并发

考虑到学生管理系统对并发要求不是特别高，暂不考虑，将来有需求在应用层面考虑可横向扩展即可。

#### 4.2.5系统的可观测

每台服务器部署一个agent模块，其作用如下：

1. 监控各服务器上各模块进程状态，是否存活；
2. 监控各服务器资源使用情况，包括内存、CPU、硬盘使用量是否超过阈值；
3. 与监控服务程序进行心跳通信，以监控业务服务器是否宕机；
4. 抓取所在服务器上部署的模块日志，发送给监控服务程序，包括事件日志和异常日志；

监控服务程序部署在一台服务器上，作用如下：

1. 检测各服务器上的agent的心跳情况；
2. 分析agent上报的报警信息；
3. 收集agent传输的日志信息，存入本地数据库；
4. 为监控PC端提供HTTP访问服务，以实时监控系统的状态。

#### 4.2.6安全性

从 PC 至 Nginx 负载均衡该通信链路使用通信通道加密，确保网络传输环境数据的安全。内网环境各子系统间的则没必要通信通道加密。

密码的存储避免明文存储。

## 4.3 设计规范

1、各子系统使用 Spring Boot 、Spring Cloud 组件开发。

2、MySQL 使用 Innodb 存储引。

3、各子系统间的调用根据实际业务需求满足：隔离、重试、幂等、超时、补偿、熔断、限流、降级使系统具备弹性。

## 5. 质量设计

### 5.1 可测试性

业务简单，开发人员容易上手，测试难度不大。

### 5.2 可维护性

技术都是使用市面比较流行通用的技术栈，开发人员能很快进入角色。

技术组件用得不复杂，运维层工作量也不大。

### 5.3 可观测性

没有什么新的特殊组件，所以针对外包学生管理系统的监控和日志收集可纳入现有标准运维体系中。

### 5.4 成本

业务系统比较简单，研发人员实现成本低，测试投入也不会太高，运维相关可纳入现有运维体系，投入不高。

硬件成本：6台服务器，其中2台Nginx、 2 台数据库。2 台业务服务器。

## ****6. 演进规划****

### 6.1 外包学生管理系统一期

需要完成所有业务功能。

需要完成系统的可观测性：进程存活状态、服务器是否宕机、日志收集。

### 6.2 外包学生管理系统二期

着重去完善系统中的弹性能力：熔断、限流、降级。

完善测试的可观测性：服务器资源监控。