2

**骨干考核系统**

**项目计划书**

团队：路人甲 汇报时间：2018-11-21

**软通动力技术服务有限公司**

目录

[概述：项目概述 3](#_Toc530574765)

[1 系统功能 3](#_Toc530574766)

[1.1 基本信息管理 3](#_Toc530574767)

[1.2 培养计划管理 3](#_Toc530574768)

[1.3 打分管理 3](#_Toc530574769)

[1.4 评价管理 3](#_Toc530574770)

[1.5 出入骨干池管理 3](#_Toc530574771)

[1.6 批量信息录入 4](#_Toc530574772)

[1.7 骨干明细查看 4](#_Toc530574773)

[1.8 权限管理 4](#_Toc530574774)

[2 系统规划 4](#_Toc530574775)

[2.1 技术方案及系统分析 4](#_Toc530574776)

[2.1.1 系统层次架构 5](#_Toc530574777)

[2.1.2 系统技术选型 5](#_Toc530574778)

[2.1.3 系统功能架构 10](#_Toc530574779)

[2.1.4 系统数据库设计与实现 10](#_Toc530574780)

[2.2 系统部署 10](#_Toc530574781)

[2.2.1 运行环境 10](#_Toc530574782)

[2.2.2系统网络拓扑 10](#_Toc530574783)

[3研发团队主要成员介绍 11](#_Toc530574784)

[4项目实施计划 11](#_Toc530574785)

# 概述：项目概述

**项目名称：**骨干考核系统

**项目背景：**主要目标是对项目成员的信息管理

# 1 系统功能

## 1.1 基本信息管理

学员基本信息，学员学分情况等基本信息管理管理。

## 1.2 培养计划管理

实现培养计划，分值等所有加分减分项目管理。

## 1.3 打分管理

实现学员每项考核具体打分，分值作为是否骨干主要依据。

## 1.4 评价管理

实现导师、HEAD、客户等关键角色对学员评价。

## 1.5 出入骨干池管理

更新最终学员是否成为骨干。

## 1.6 批量信息录入

实现骨干相关信息的批量录入，提高使用者工作效率。

## 1.7 骨干明细查看

可按照不同维度查看骨干数量、分布及详情。

## 1.8 权限管理

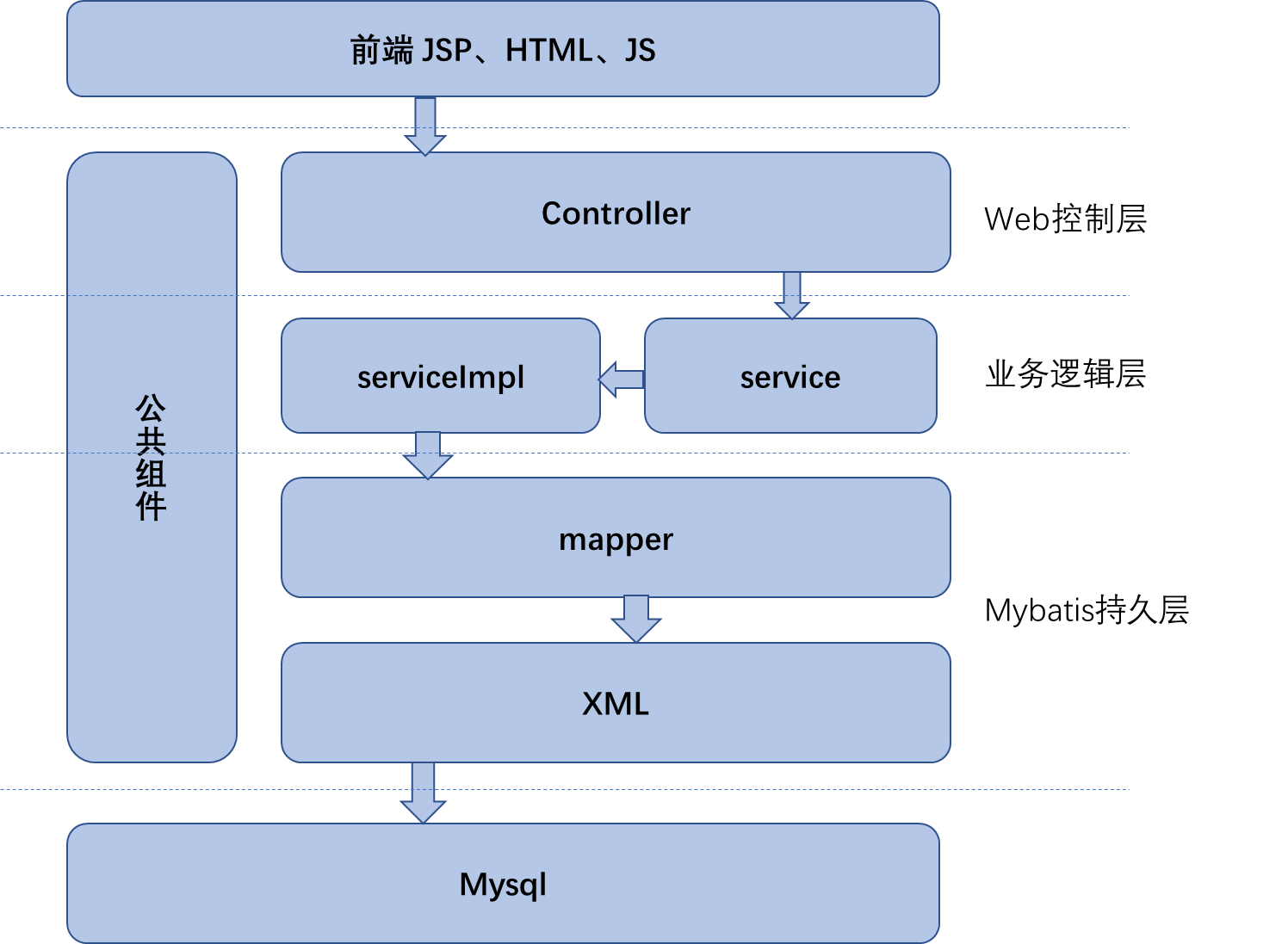
系统区分管理员，信息录入员和普通用户三种权限

# 2 系统规划

## 2.1 技术方案及系统分析

系统采用用前后端分离模式开发，后台使用spring boot + mybatis +SQLServer实现restful接口，前台使用VUE + elementUi框架。

### 2.1.1 系统层次架构



### 2.1.2 系统技术选型

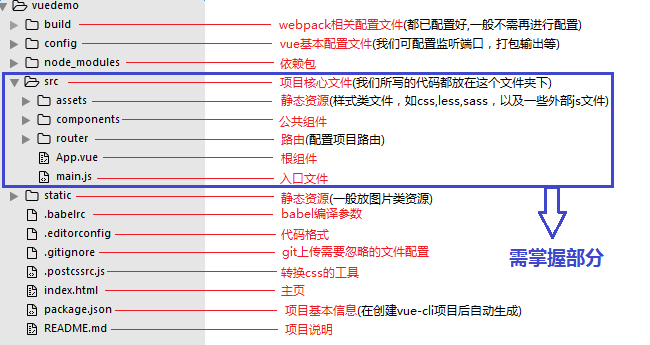
#### 2.2.2.2 前台框架技术细节

综合考虑，项目场景、团队技能、以及开发周期情况，进行前端技术选型：

|  |  |  |
| --- | --- | --- |
| **技术** | **名称** | **版本** |
| Vue | 基础框架 | 2.0 |
| Element UI | 组件库 | 2.4.10 |

**A、VUE代码目录结构**

一个vue-cli的项目结构如下，其中src文件夹是需要掌握的，所以本文也重点讲解其中的文件，至于其他相关文件，了解一下即可。



**B、Element UI：完整的组件列表**



#### 2.2.2.1 后台框架技术细节

综合考虑，项目场景、团队技能、以及开发周期情况，进行后台技术选型：

|  |  |  |
| --- | --- | --- |
| **技术** | **名称** | **版本** |
| jdk | Java开发环境 | 8u144（64位） |
| spring boot | 基础框架 | 2.0.6 |
| MyBatis | ORM框架 | 1.1.1 |
| PageHelper | MyBatis物理分页插件 | 5.1.2 |
| Apache shiro | 安全框架 | 1.4.0 |
| Druid | 数据库连接池 | 2.0 |
| Swagger2 | 接口测试框架 | 2.0 |
| Quartz | 作业调度框架 | 2.1.1 |
| Maven | 项目构建管理 | 4.0.0 |
| mysql | 数据库 | 5.6 |

**A、JDK**

java8，性能、稳定、内存优化

**B、****Springboot**

* 遵循“习惯优于配置”的原则，使用Spring Boot只需要很少的配置，大部分的时候我们直接使用默认的配置即可；
* 项目快速搭建，可以无需配置的自动整合第三方的框架；
* 可以完全不使用XML配置文件，只需要自动配置和Java Config；
* 内嵌Servlet容器，降低了对环境的要求，可以使用命令直接执行项目，应用可用jar包执行：java -jar；
* 提供了starter POM, 能够非常方便的进行包管理, 很大程度上减少了jar hell或者dependency hell；
* 运行中应用状态的监控；
* 对主流开发框架的无配置集成；
* 可极大地提高开发、部署效率；

**C、ORM选型（MyBatis）**

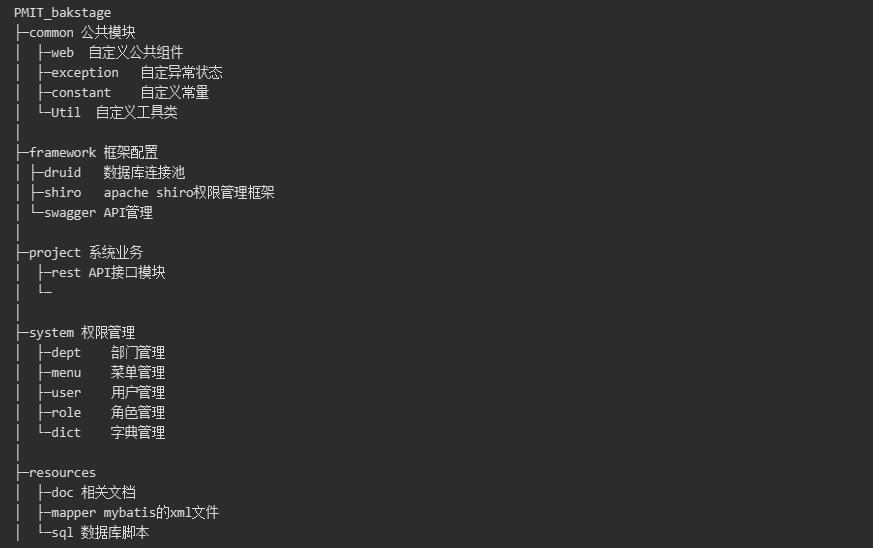
* 不屏蔽SQL ,意味着可以更精确地定位sql语句，可以对其进行优化和改造，这有利于互联网系统性能的提高，符合互联网需要性能优化的特点。
* 提供强大、灵活的映射机制，方便java开发者使用。提供动态的sql的功能，允许我们根据不同条件组装sql，这个功能远比其他工具或java编码的可读性和可维护性高的多，满足各种应用系统的同时也满足了需求经常变化的互联网应用的要求。
* 在mybatis中，提供使用了Mapper接口编程，只要一个接口和一个XML就能创建映射器，进一步简化我们的工作，使得很多框架API在mybatis中消失，开发者更集中于业务逻辑。

**D、权限管理选型（shiro）**

Shiro是一个强大而灵活的开源安全框架，能够非常清晰的处理认证、授权、管理会话以及密码加密。如下是它所具有的特点：

* 易于理解的 Java Security API；
* 简单的身份认证（登录），支持多种数据源（LDAP，JDBC，Kerberos，ActiveDirectory 等）；
* 对角色的简单的签权（访问控制），支持细粒度的签权；
* 支持一级缓存，以提升应用程序的性能；
* 内置的基于 POJO 企业会话管理，适用于 Web 以及非 Web 的环境；
* 异构客户端会话访问；
* 非常简单的加密 API；
* 不跟任何的框架或者容器捆绑，可以独立运行。

**E、代码目录结构**



### 2.1.3 系统功能架构

### 2.1.4 系统数据库设计与实现

## 2.2 系统部署

### 2.2.1 运行环境

**客户端**

　　操作系统：Windows XP以上

　　浏览器：Microsoft Internet Explore 8.0以上

**服务器端**

　　操作系统：Windows 7

　　数据库： MySQL 5.6

Web服务：apache-tomcat-7.0.55

jre：jre-8u144-windows-x64

### 2.2.2系统网络拓扑

 可通过timesheet申请反向代理，实现黄区网络访问。

# 3 研发团队主要成员介绍



# 4项目实施计划

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 阶段 | 描述 | 开始日期 | 结束日期 |
| 1 | 需求分析 | 根据调研结果确定项目需求，输出需求文档及规格文档 | 2018-11-16 | 2018-11-22 |
| 2 | 架构设计 | 进行架构设计，确定系统架构框架、内部/外部接口；物料准备 | 2018-11-22 | 2018-11-23 |
| 3 | 原型设计 | 实现系统的原型及基本功能 | 2018-11-22 | 2018-11-23 |
| 4 | 迭代开发 | 详细设计、编码、单元测试、UI | 2018-11-27 | 2018-11-30 |
| 5 | 内部测试 | 内部联调 | 2018-12-1 | 2018-12-7 |
| 6 | Showcase | 最终问题收编，结果展示 | 2018-12-8 | 2018-12-12 |
| 总计 |  |  |  |  |

Ps：附图

