



项目名称： 基于 SpringBoot 和 face-recognition 的
互联网健身监管系统

项目组长： 胡悦

项目成员： 陈泽平、胡悦、穆晓丹、任小健、夏兆喆、杨嘉欣

指导老师： 赵斌

关 键 词： SpringBoot、SSM、face-recognition、Vue.js

二〇一八年九月三十日

目 录

第 1 章	项目背景.....	3
第 2 章	开发计划.....	4
2.1	功能性需求	4
2.2	非功能性需求	8
2.3	设计约束.....	10
2.3.1	SSM	10
2.3.2	Spring Boot	10
2.3.3	Vue.js.....	10
2.3.4	face-recognition	11
第 3 章	系统设计	12
第 4 章	后台设计.....	13
第 5 章	前端设计	14

第1章 项目背景

健身行业作为一个热门的朝阳产业，具有较大的发展潜力。同时，因其辐射范围的局限性，市场垄断可能性小，从而拥有一定的投资市场空间。相较于传统模式的健身房，互联网式健身房具有租赁成本、人力成本低的优势。这使得互联网式健身房可推出低价月卡、免费预约团课这一独特且具有较好短期回报率的会员制度，从而令顾客的决策成本降低，提高健身平台的获客率及顾客的留存率。

目前整个健身市场规模没有统一数据，但从多方调研的结果可得知市场大致规模和营业额。根据不同业内人士统计和计算，以大约 1500 万国内持健身卡人数计算，大致可以得出在 2015 年，全国成规模的健身房大概 1 万家，总体市场规模在 300 亿元的结论。

与发达国家相比，我国健身市场远未达到天花板。据报告，2015 年中国线下健身会员仅有 350 万人，而美国有 5,020 万人，中国的人均教练只有 0.12 人/万人，美国则达 7.5 人/万人。中国健身人群每年健身投入费用为 1 万元左右，商业健身市场规模为 300 亿人民币，而 2013 年美国的健身市场规模即达 218 亿美元。即使与亚太地区相比，中国的健身会员渗透率仅为 0.4%，仅高于印度和印尼，显著低于亚洲平均水平 3.8%。随着生活水平的提升以及消费观念的改变，中国城乡居民更加重视健康，尤其是青少年的身体素质，对体育运动的参与度也在提高。2015 年我国经常参与体育锻炼的人口约为 4 亿，体育产业规模为 1.35 万亿

目前国内 SaaS 健身房管理系统稀缺，功能上存在的差异也较大，据观察大部分的差异体现在社交营销这一块。“互联网+”代表一种新的经济形态，即充分发挥互联网在生产要素配置中的优化和集成作用，将互联网的创新成果深度融合于经济社会各领域之中，提升实体经济的创新力和生产力，形成更广泛的以互联网为基础设施和实现工具的经济发展新形态。

“互联网+”行动计划将重点促进以云计算、物联网、大数据为代表的新一代信息技术与现代制造业、生产性服务业等的融合创新，发展壮大新兴业态，打造新的产业增长点，为大众创业、万众创新提供环境，为产业智能化提供支撑，增强新的经济发展动力，促进国民经济提质增效升级。通俗来说，“互联网+”就是“互联网+各个传统行业”，但这并不是简单的两者相加，而是利用信息通信技术以及互联网平台，让互联网与传统行业进行深度融合，创造新的发展生态。这相当于给传统行业加一双“互联网”的翅膀，然后助飞传统行业。

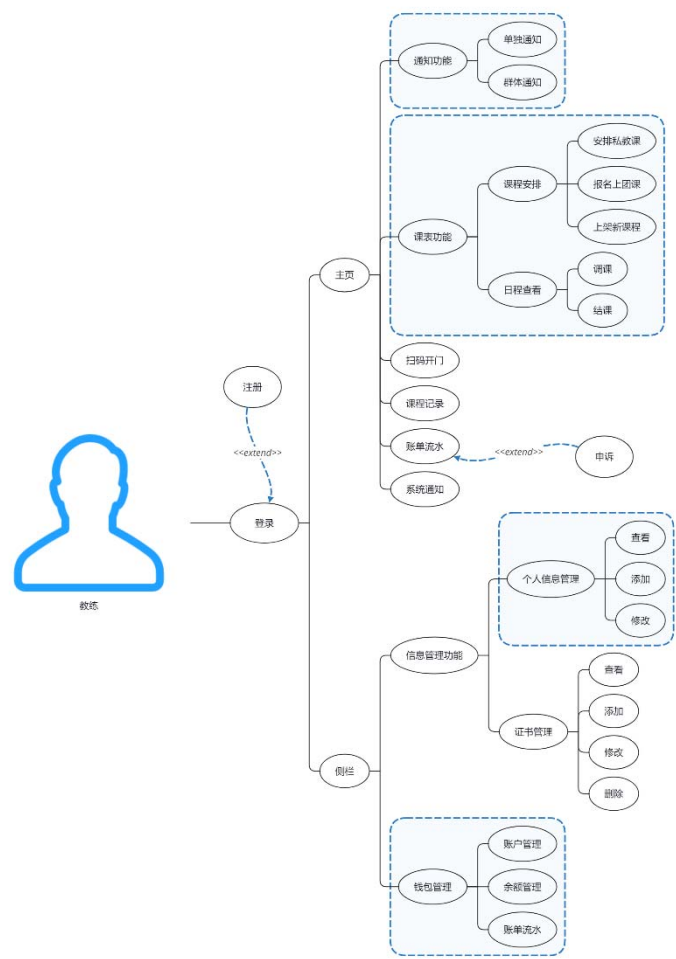
本项目的成立是一种顺应国情的表现，国内健身行业发展不比国外，尽管在国内健身房是朝阳产业，但健身氛围比国外实在差得太多，销售意味太过浓厚。所以系统在原有基础上做了优化和调整顺应发展趋势。

第2章 开发计划

2.1 功能性需求

教练端功能实现：

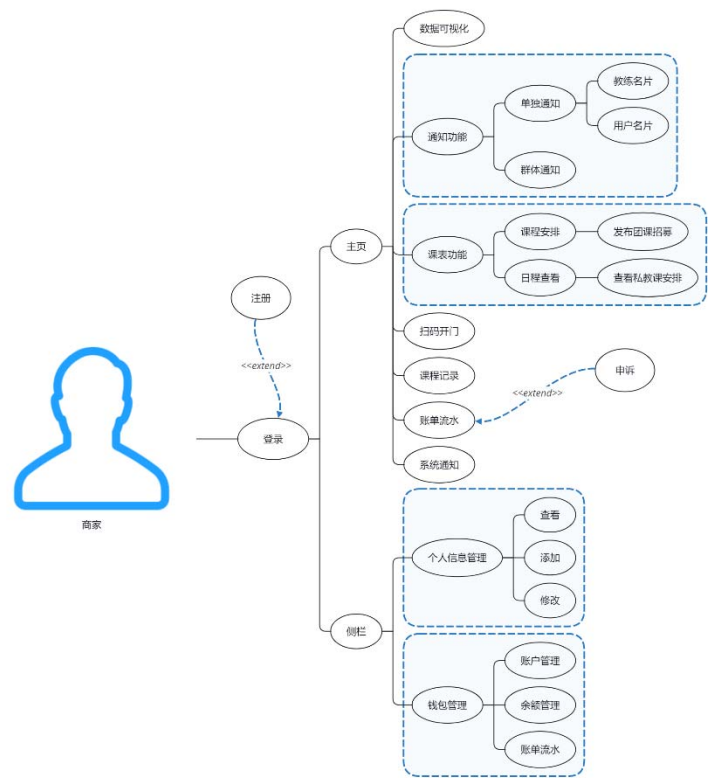
- 注册登录功能
- 扫码开门功能
- 通讯录功能
- 个人信息管理功能
- 钱包功能



图表 2-1 教练端需求分析用例图

商家端功能实现：

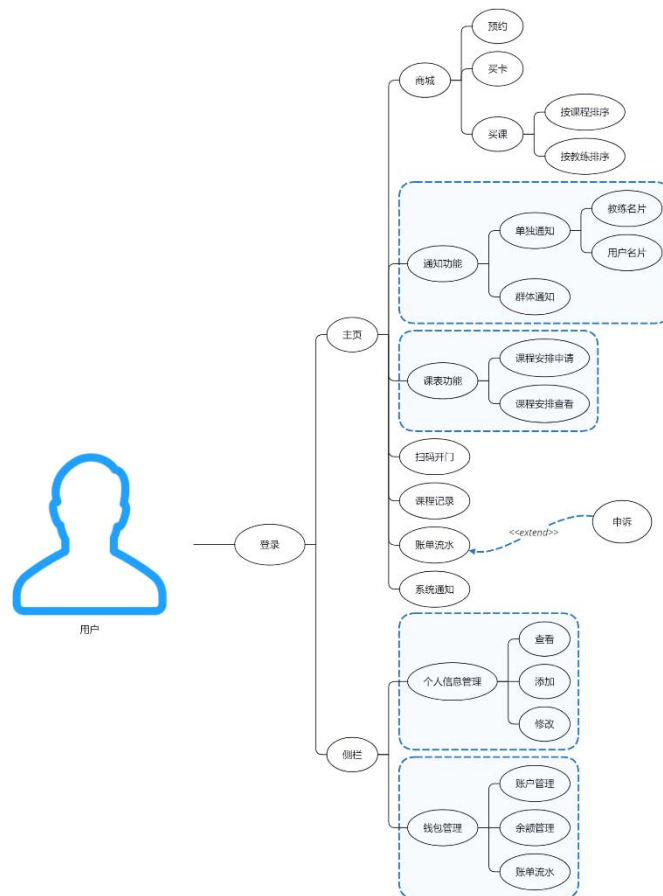
- 注册登录功能
- 扫码开门功能
- 通讯录功能
- 数据可视化功能
- 个人信息管理功能
- 钱包功能



图表 2-2 商家端需求分析用例图

用户端功能实现：

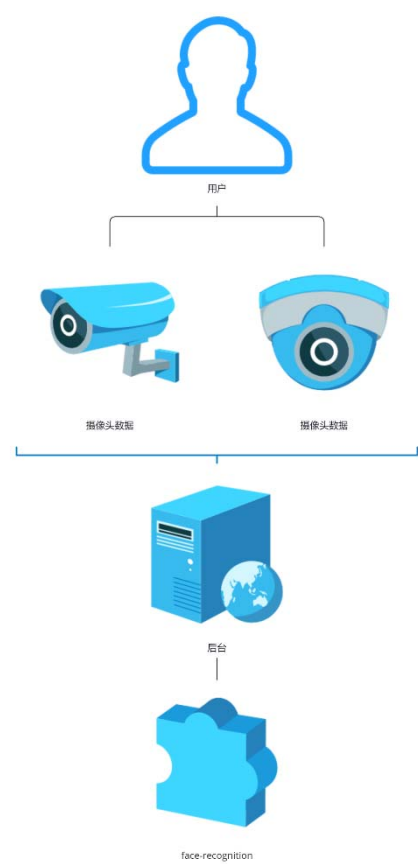
- 登录注册功能
- 扫码开门功能
- 商城功能
- 个人信息管理功能
- 钱包功能



图表 2-3 用户端需求分析用例图

监管功能实现：

- 人脸识别
- 感应抓拍



图表 2-4 监管功能需求分析图

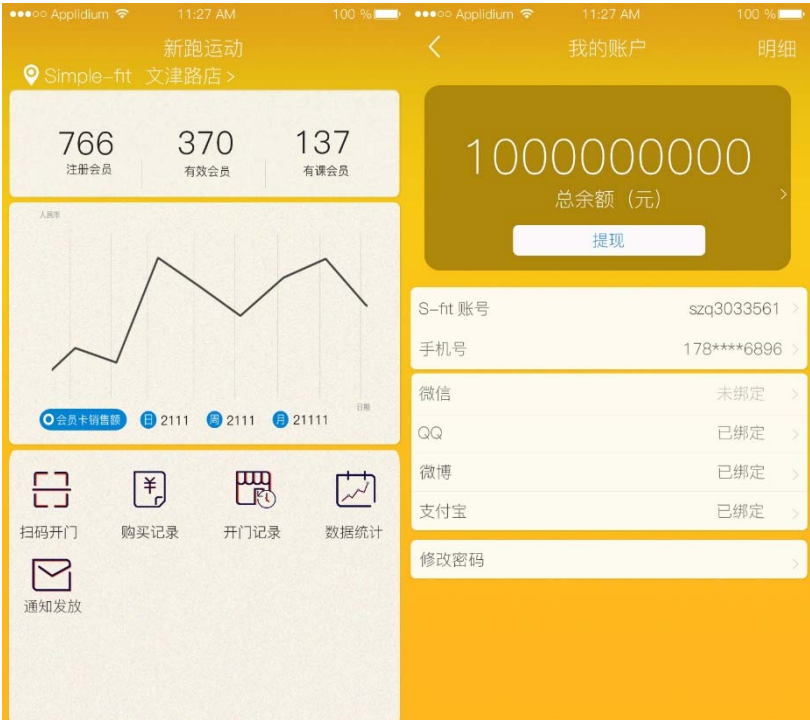
2.2 非功能性需求

教练端 UI 设计:



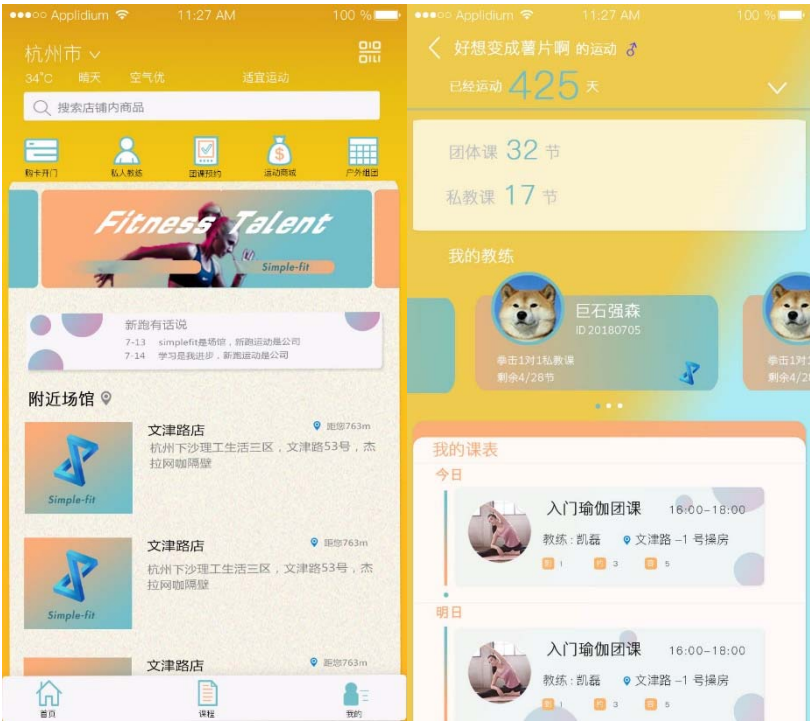
图表 2-5 教练端 UI 设计图

商家端 UI 设计:

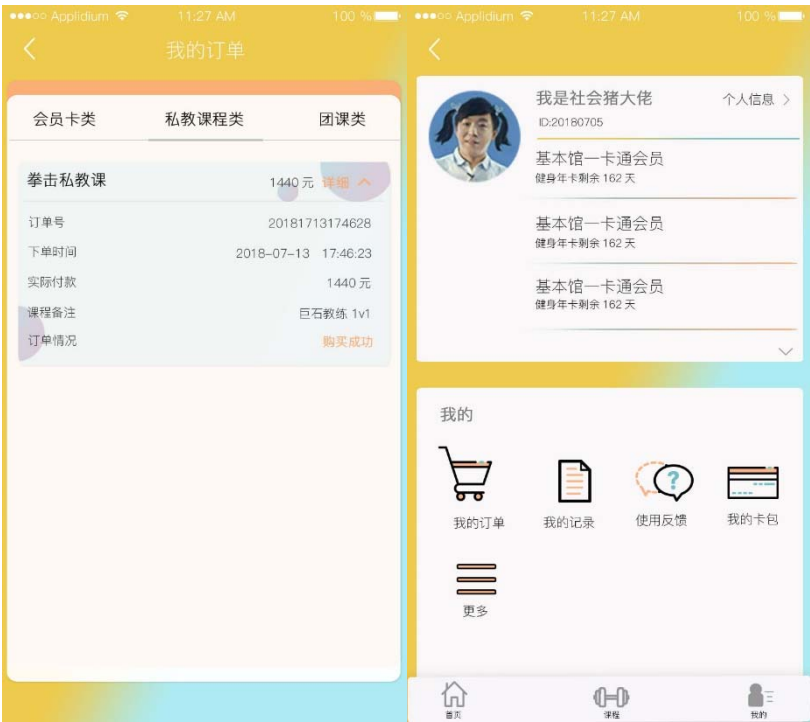


图表 2-6 商家端 UI 设计图

用户端 UI 设计：



图表 2-7 用户端 UI 设计图 1



图表 2-8 用户端 UI 设计图 2

2.3 设计约束

2.3.1 SSM

SSM(Spring+SpringMVC+MyBatis)框架集由 Spring、MyBatis 两个个开源框架整合而成(SpringMVC 是 Spring 中的部分内容)。常作为数据源较简单的 web 项目的框架。其中 Spring 是一个轻量级的控制反转 (IoC) 和面向切面 (AOP) 的容器框架。SpringMVC 分离了控制器、模型对象、分派器以及处理程序对象的角色, 这种分离让它们更容易进行定制。MyBatis 是一个支持普通 SQL 查询, 存储过程和高级映射的优秀持久层框架。页面发送请求给控制器, 控制器调用业务层处理逻辑, 逻辑层向持久层发送请求, 持久层与数据库交互, 后将结果返回给业务层, 业务层将处理逻辑发送给控制器, 控制器再调用视图展现数据。

2.3.2 Spring Boot

Spring Boot 是由 Pivotal 团队提供的全新框架, 其设计目的是用来简化新 Spring 应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置, 从而使开发人员不再需要定义样板化的配置。通过这种方式, Spring Boot 致力于在蓬勃发展的快速应用开发领域(rapid application development)成为领导者。

从最根本上来讲, Spring Boot 就是一些库的集合, 它能够被任意项目的构建系统所使用。简便起见, 该框架也提供了命令行界面, 它可以用来运行和测试 Boot 应用。框架的发布版本, 包括集成的 CLI(命令行界面), 可以在 Spring 仓库中手动下载和安装。一种更为简便的方式是使用 Groovy 环境管理器(Groovy Environment Manager, GVM), 它会处理 Boot 版本的安装和管理。Boot 及其 CLI 可以通过 GVM 的命令行 `gvm install springboot` 进行安装。在 OS X 上安装 Boot 可以使用 Homebrew 包管理器。为了完成安装, 首先要使用 `brew tap pivotal/tap` 切换到 Pivotal 仓库中, 然后执行 `brew install springboot` 命令。

要进行打包和分发的工程会依赖于像 Maven 或 Gradle 这样的构建系统。为了简化依赖图, Boot 的功能是模块化的, 通过导入 Boot 所谓的“starter”模块, 可以将许多的依赖添加到工程之中。为了更容易地管理依赖版本和使用默认配置, 框架提供了一个 parent POM, 工程可以继承它。

2.3.3 Vue.js

Vue.js 是一套构建用户界面的渐进式框架。与其他重量级框架不同的是, Vue 采用自底向上增量开发的设计。Vue 的核心库只关注视图层, 并且非常容易学习, 非常容易与其它库或已有项目整合。另一方面, Vue 完全有能力驱动采用单文件组件和 Vue 生态系统支持的库开发的复杂单页应用。Vue.js 的目标是通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件。Vue.js 自身不是一个全能框架——它只聚焦于视图层。因此它非常容易学习, 非常容易与其它库或已有项目整合。另一方面, 在与相关工具和支持库一起使用时, Vue.js 也能完美地驱动复杂的单页应用。

2.3.4 face-recognition

人脸识别大致分为如下三个部分：人脸检测(face detection)、人脸对齐(face alignment)、人脸识别(face recognition)。其中人脸识别包括 face verification 即两张图片相似程度和 face identification 在图片库中检索与当前图片相似度最高的图片。

人脸检测(face detection)通常认为是在图片中找到人脸的过程，这个过程常用的办法有用 openCV 调用 'haarcascade_frontalface_default.xml' 或者其他 xml 文件构建分类器对象，通过 detectMultiScale 函数返回人脸框坐标，在图像中框出人脸。也可以通过深度学习的方法定位人脸。

人脸对齐(face alignment)指在标定人脸位置后对人脸上的特征进一步定位，可以对人脸检测的侧脸进行校正、旋转、3d 变换等，也可以对特征点进行特定的表情变化。常用的有 5 点和 68 点特征。如 DCNN，TDCNN，MTCNN 等方法。

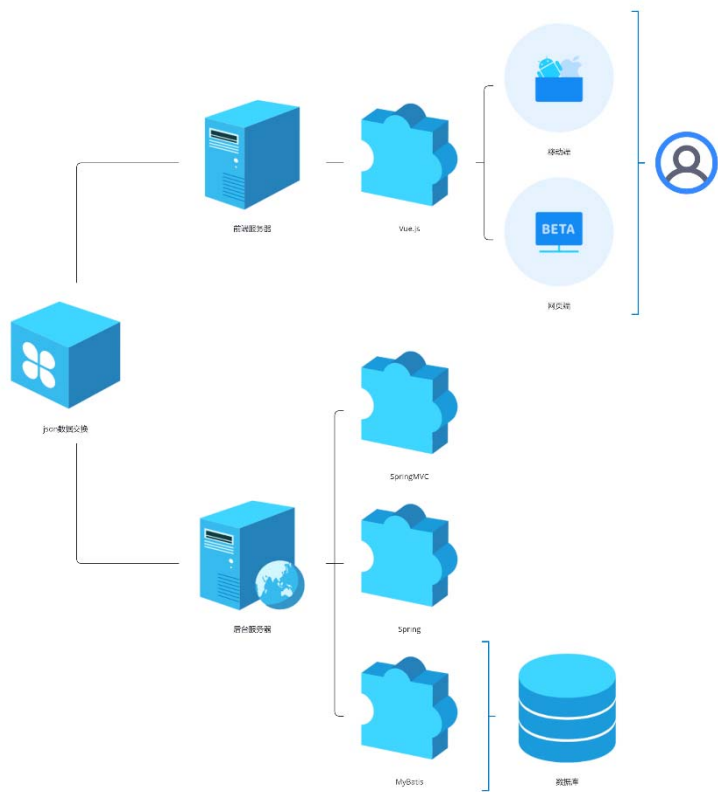
人脸识别(face recognition)，人脸识别是由 CNN 提取人脸的特征，将两张待测图片输入至训练好的 CNN，提取全连接层的特征。模型一般是由预训练的分类模型演变，如 1000 个人，每个人 50 张图片训练的 CNN 模型。face verification 两张待测图片在训练好的模型上提取全连接层特征，如 vgg 的 fc7 层的 4096 维特征，进行 cosin 距离计算。利用 vgg-facecaffemodel，如果角度光照相同，可以达到 0.93 左右的准确率，戴眼镜的影响很大。



图表 2-9 face-recognition 网络概念图

第3章 系统设计

系统选择前后端分离的开发模式，在以前传统的网站开发中，前端一般扮演的只是切图的工作，只是简单地将 UI 设计师提供的原型图实现成静态的 HTML 页面，而具体的页面交互逻辑，比如与后台的数据交互工作等，可能都是由后台的开发人员来实现的，或者是前端是紧紧的耦合后台。比如，以前淘宝的 Web 基本上都是基于 MVC 框架 webx，架构决定了前端只能依赖后端。所以他们的开发模式依然是，前端写好静态 demo，后端翻译成 VM 模版，这种模式的问题很多。而且更有可能后台人员直接兼顾前端的工作，一边实现 API 接口，一边开发页面，两者互相切换着做，而且根据不同的 url 动态拼接页面，这也导致后台的开发压力大大增加。前后端工作分配不均。不仅仅开发效率慢，而且代码难以维护。而前后端分离的话，则可以很好的解决前后端分工不均的问题，将更多的交互逻辑分配给前端来处理，而后端则可以专注于其本职工作，比如提供 API 接口，进行权限控制以及进行运算工作。而前端开发人员则可以利用 nodejs 来搭建自己的本地服务器，直接在本地开发，然后通过一些插件来将 api 请求转发到后台，这样就可以完全模拟线上的场景，并且与后台解耦。前端可以独立完成与用户交互的整一个过程，两者都可以同时开工，不互相依赖，开发效率更快，而且分工比较均衡。



图表 3-1 系统设计图

第4章 后台设计

SSM 框架是指：Spring+Spring MVC+Mybatis。

- Spring

轻量：从大小与开销两方面而言 Spring 都是轻量的，此外，Spring 是非侵入式的：典型地，Spring 应用中的对象不依赖于 Spring 的特定类。简单说就是自己需要什么就导入对应的 jar 的即可，而不是侵入式的。

控制反转：Spring 通过一种称作控制反转（IoC）的技术促进了低耦合，当应用了 IoC，一个对象依赖的其它对象会通过被动的方式传递进来，而不是这个对象自己创建或者查找依赖对象。

面向切面：Spring 提供了面向切面编程的丰富支持，允许通过分离应用的业务逻辑与系统级服务（例如审计（auditing）和事务（transaction）管理）进行内聚性的开发。应用对象只实现它们应该做的——完成业务逻辑——仅此而已。它们并不负责（甚至是意识）其它的系统级关注点，例如日志或事务支持。

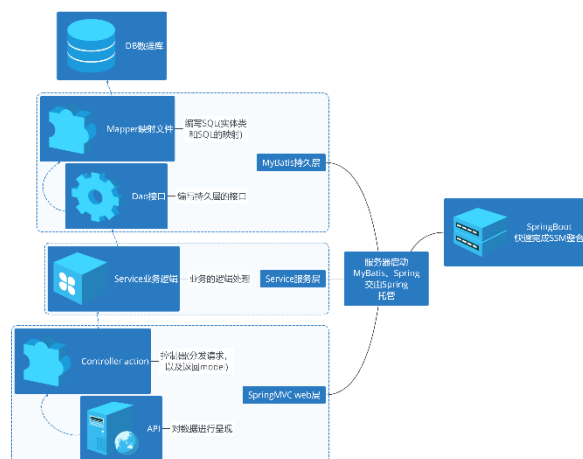
框架：Spring 可以将简单的组件配置、第三方框架组合成为复杂的应用。

- Spring MVC

MVC 框架是一个全功能的构建 Web 应用程序的 MVC 实现。通过策略接口，MVC 框架变成高度可配置的，MVC 容纳了大量视图技术，其中包括 JSP、Velocity、Tiles、iText 和 POI。模型由 javaBean 构成，存放于 Map；视图是一个接口，负责显示模型；控制器表示逻辑代码，是 Controller 的实现。

- Mybatis

MyBatis 是一个基于 Java 的持久层框架。MyBatis 消除了几乎所有的 JDBC 代码和参数的手工设置以及结果集的检索。MyBatis 使用简单的 XML 或注解用于配置和原始映射，将接口和 Java 的 POJOs（普通的 Java 对象）映射成数据库中的记录。



图表 4-1 后台设计图

- SpringBoot

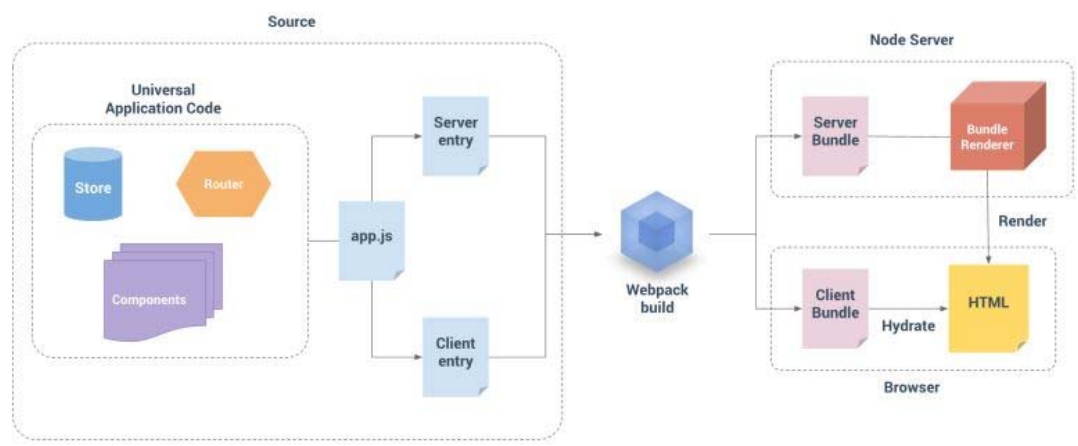
SpringBoot 已经是大势所趋，在此之前项目在进行整合 SSM 的时候要花好久的时间，而且还容易出错，大量的 xml 配置，每个框架的版本必须相对应，不然整合后会出现各种各样的问题。SpringBoot 能快速完成 SSM 的整合，大大提高了开发效率。

第5章 前端设计

Vue.js 是一个 JavaScriptMVVM 库，是一套构建用户界面的渐进式框架。它是以数据驱动和组件化的思想构建的，采用自底向上增量开发的设计。相比于 Angular.js，Vue.js 提供了更加简洁、更易于理解的 API，使得我们能够快速地上手并使用 Vue.js。

在 vue 调试方面，可以选择安装 chrome 插件 vue Devtools。打开 vue 项目，在 console 控制台选择 vue 面板。在 Devtools 工具中，可以选择组件，查看对应组件内的数据信息。也可以选择 Vuex 选项，查看该项目内 Vuex 的状态变量信息。

在 vue 组件库方面，更重视开发效率，并且选择了 vue2.0 作为前端框架，那么饿了么推出的 Element 组件就是一个很不错的选择。其 github 项目 (<https://github.com/ElmeFE/element>)更新比较频繁，虽然项目会有些不稳定，但是目前为止 element 就是最好的支持 vue2.0 的 UI 组件。就像它的口号一样，“快速成型，就为让你少加班”。



图表 5-1 前端设计图