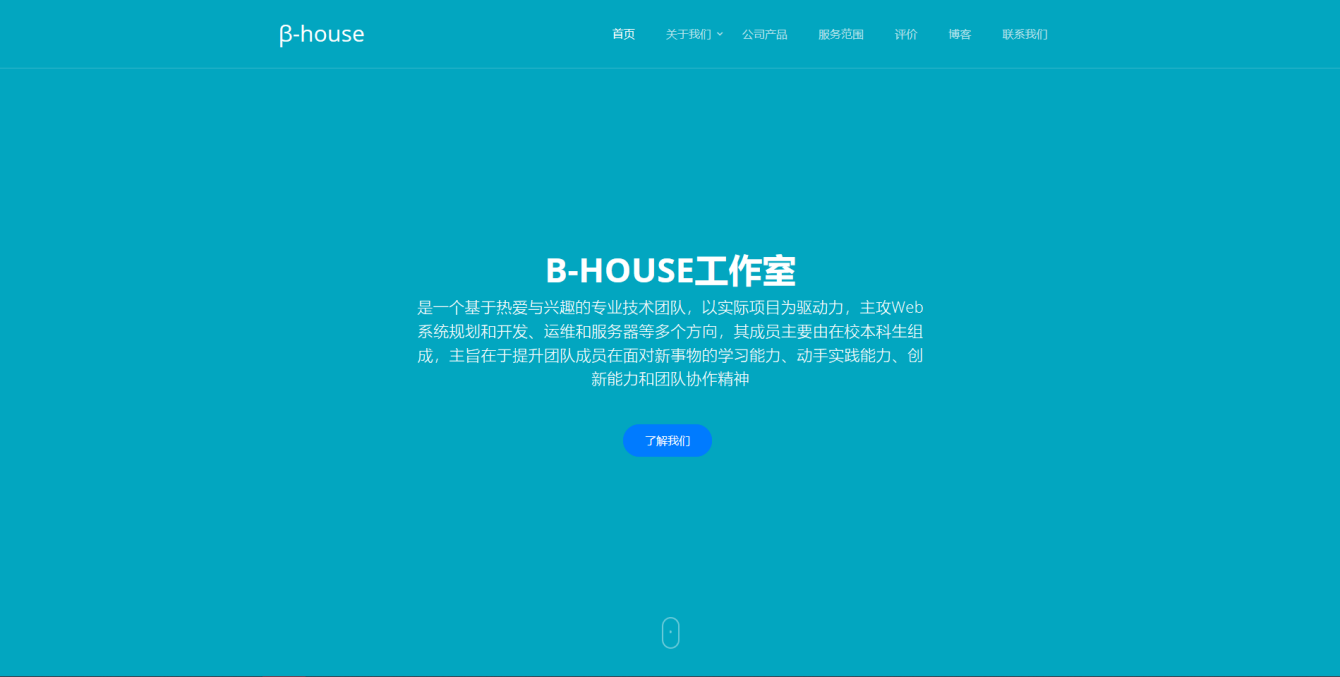
**第一部分 β-house公司介绍**

**1.1 β-house公司介绍:**

公司网页及相关在线WEBAPP服务支持

为了让用户能够更好的了解公司概况以及产品信息，我们建立了公司官网来实时发布公司相关概况，以及关于“第二课堂”产品的使用说明和相关教程。除此之外，考虑到产品后期推广,根据不同学校不同需求,我们提供的网页端，能够使用户可以通过浏览器访问公司主页来获得相关服务支持。

**1.2主页展示**

公司网页主要分为关于我们、公司产品、服务范围、联系我们这四大板块。在这里，用户不仅能够及时地查看最新的产品信息，公司信息，获取前沿资讯，而且能够查阅相关的教程，了解相关产品的使用方法，除此之外，用户还能在这里获取类似APP之类等辅助软件的下载方式。一旦使用出现相关问题，客户可以及时的通过网页联系我们团队，通过实时跟进与沟通了解，得到满意的解决方案。

### 1.1.1“公司信息”展示

在公司信息板块中，可以查看公司信息简介、企业文化、核心价值观等内容，能够使用户对我们β-house有限责任公司一个更清晰直观的了解。

β-house有限责任公司，是一家基于网站平台及微信平台构建，利用React,Vue等新兴技术广泛融合，结合社会需求，以应对人工记录效率低,用户作弊手段多的服务型公司，我们提供专业化、人性化的服务，促进社会和谐稳定发展。

本公司致力于建设一个高效、规范、温暖、技术强的团队，我们相信竞争与合作的魅力，我们坚守团结与务实的信条，我们崇尚技术改变生活，我们希望通过团队的发展推动个人的自我超越，同时促进我们服务质量的提升，达到消费者、投资者、成员以及社会的平衡。



图 1-1-1 企业核心价值观展示

**公司的宗旨**：效率成就品牌，创新铸就未来

**公司的服务口号是**：

您之所想，我之所思

以年轻的思维、细心、技术

创造全新体验,享受便捷生活

**公司的目标**：

1.技术层面上，公司关注前沿技术,并争取将前沿技术加入软件设计中,争取为用户提供最优体验。

2.社会层面上，产品为社会服务,为人类服务,本公司专注引领电子化记录普及.践行低碳绿色,为社会贡献一份力量.

3.经济层面上，作为一家公司最重要之一的便是公司的业绩和所得到的利润。我们希望在两年内总体实现盈利，并且为员工薪酬做出很好的保障。

4.顾客需求上，顾客作为我们的上帝，本公司愿用科学的方法满足他们所提出的的要求，并且赢得客户的信赖和支持。

### 1.1.2 “产品信息”展示



图 1-1-2 产品信息展示

本项目使用前沿软件设计技术，形成我们的核心产品——”第二课堂”，在产品信息中可以看到相应的介绍，以及产品的主要功能特点，还配备相关使用介绍和教程，如果有任何疑难问题，都可以向我们咨询。

### 1.1.3 “服务支持”展示

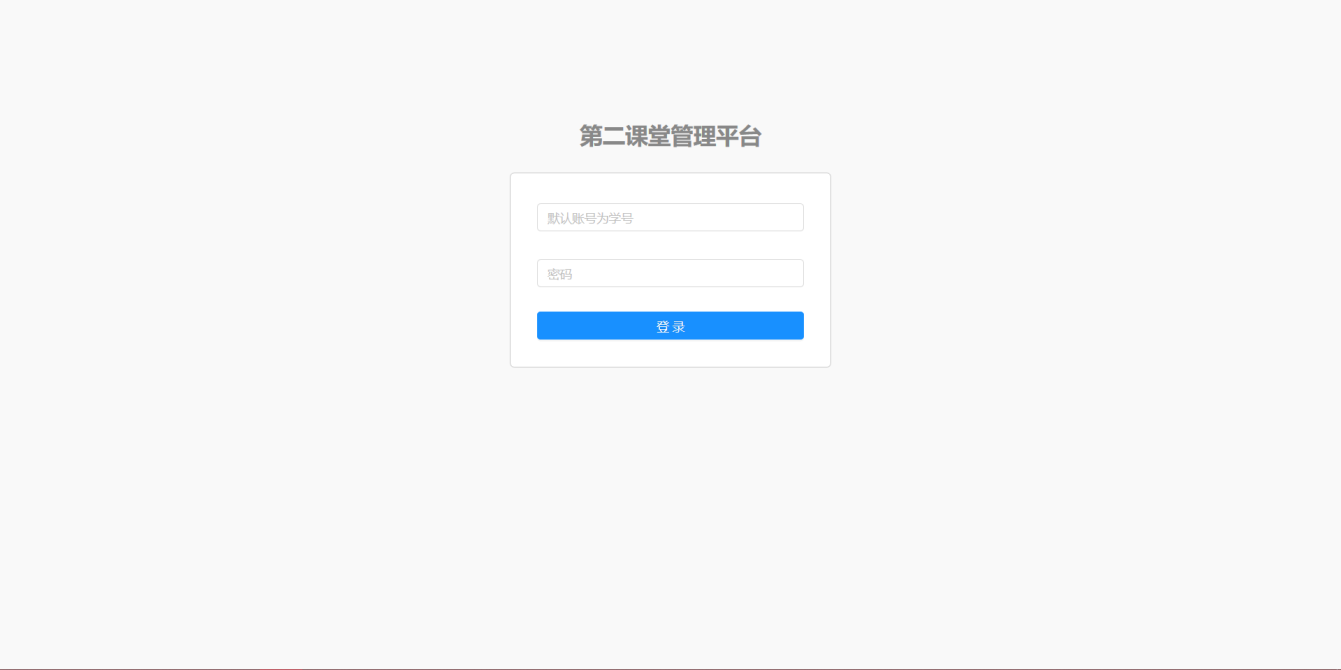


图 1-1-3-1服务支持展示

在简单的数据交互后，我们对购买了相应产品的用户开放相应的服务支持权限，我们只会对经过严格检验后的，拥有该信息查询权力的用户开放相应的服务查询权限，确保登录的用户是享有相应权力的用户，保证相关的私人数据不被轻易泄露。

我们还绘制出详细的各种数据图表，让用户能够更好的去了解并分析数据，简单的数据例如单日盖章数目,物资借用记录趋势图,小程序日活跃人数统计等功能。

以下简单的展示一些登录用户后能够查看的一些数据画面。

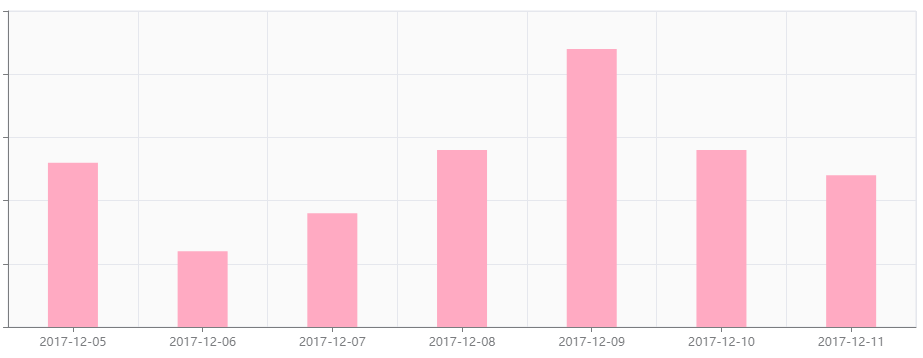


图 1-1-3-2 在线直方图每日盖章统计

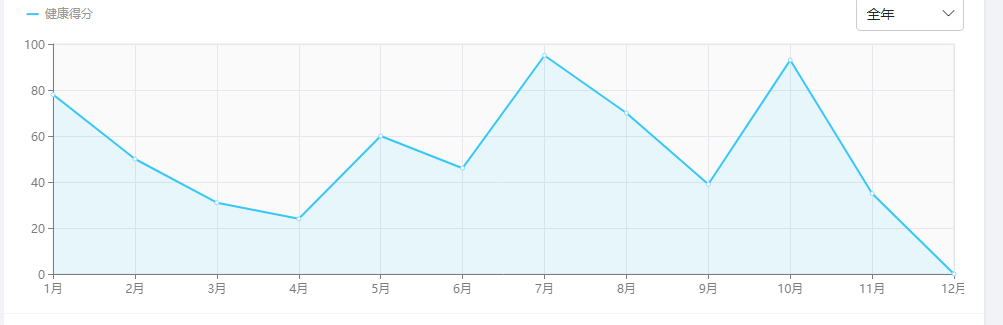


图 1-1-3-3 网页安全性评估

根据反馈的数据，用户可以简单且及时地查看单日盖章数目，一旦发生异常-----盖章数目大于预估盖章数目,用户可及时查看系统日志,进行错误查找,防止一些问题由于早期显现的不明显而被忽略。

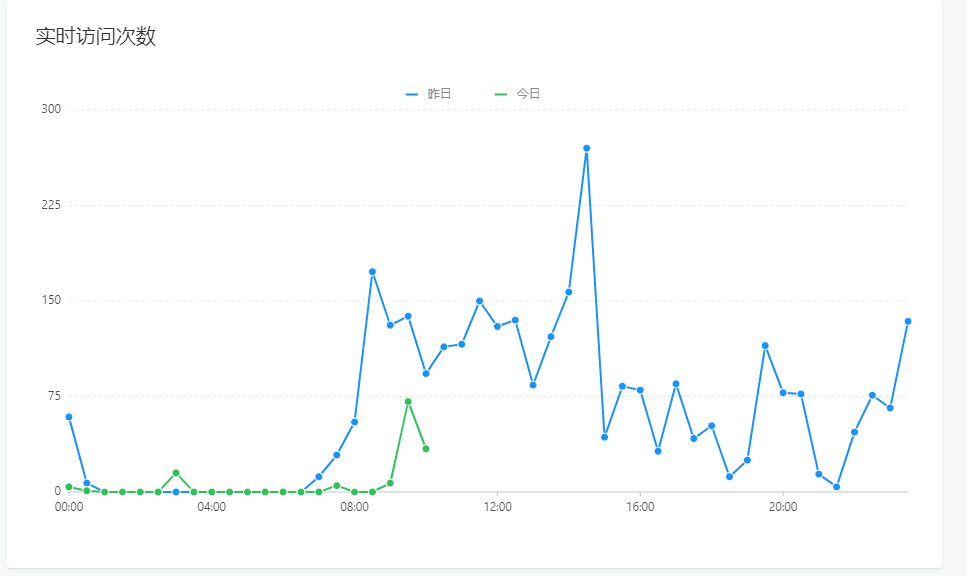


图1-1-3-4 实时访问人数

### 1.1.4 “联系我们”展示

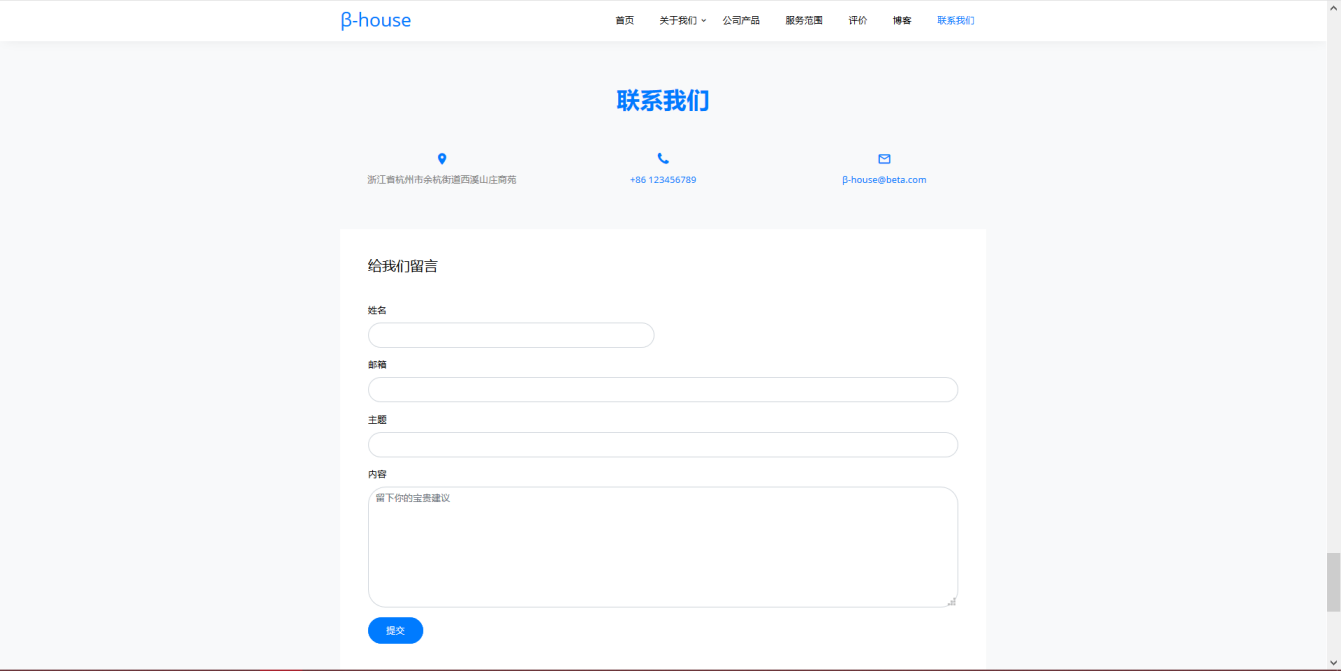


图 1-1-4联系我们展示

“联系我们”这一板块的设置主要是给出了我们的办公室地址、联系电话、联系邮箱等信息，用户碰到有任何问题都可以通过电话、邮箱等方式与我们取得联系，β-house有限责任公司24小时竭诚为您服务！！！

**第二部分 第二课堂技术支持**

### **2.1前后端分离技术**

**前后端分离**

本项目开发使用了前后端分离技术，即前端专注于数据展示与交互，后端专注业务逻辑的处理。在架构上分离解耦，逐渐摆脱前后端在架构上的依赖，前后端单独开发，通过 Restful接口传递数据。减轻后端服务器的压力，后端服务器不再负责页面渲染，只负责输入数据，提升吞吐量。

**2.1.1 Git团队开发**

**git项目管理**

Git是一个开源的分布式版本控制系统，用以有效、高速的处理从很小到非常大的项目版本管理。git对源代码进行管理,使本地机器与远程服务器代码保持同步,让多人进行项目开发,对同一份代码进行更改,也可根据项目需求同步他人代码。当项目遇到未预期错误,使用git版本管理可以清晰了解不同版本之间区别,对代码进行回滚处理。





**2.1.2 Maven多模块管理**

**Apache Maven**，是一个[软件](https://zh.wikipedia.org/wiki/%E8%BD%AF%E4%BB%B6" \t "https://www.yuque.com/cpwen/pyigqn/_blank)（特别是[Java](https://zh.wikipedia.org/wiki/Java_(%E7%BC%96%E7%A8%8B%E8%AF%AD%E8%A8%80)" \t "https://www.yuque.com/cpwen/pyigqn/_blank)软件）[项目管理](https://zh.wikipedia.org/wiki/%E9%A1%B9%E7%9B%AE%E7%AE%A1%E7%90%86" \t "https://www.yuque.com/cpwen/pyigqn/_blank)及[自动构建](https://zh.wikipedia.org/wiki/%E8%87%AA%E5%8A%A8%E6%9E%84%E5%BB%BA" \t "https://www.yuque.com/cpwen/pyigqn/_blank)工具，由[Apache软件基金会](https://zh.wikipedia.org/wiki/Apache%E8%BD%AF%E4%BB%B6%E5%9F%BA%E9%87%91%E4%BC%9A" \t "https://www.yuque.com/cpwen/pyigqn/_blank)所提供。基于项目对象模型（缩写：POM）概念，Maven利用一个中央信息片断能管理一个项目的构建、报告和文档等步骤。

Maven也可被用于构建和管理各种项目，例如[C#](https://zh.wikipedia.org/wiki/C%E2%99%AF" \t "https://www.yuque.com/cpwen/pyigqn/_blank)，[Ruby](https://zh.wikipedia.org/wiki/Ruby" \t "https://www.yuque.com/cpwen/pyigqn/_blank)，[Scala](https://zh.wikipedia.org/wiki/Scala" \t "https://www.yuque.com/cpwen/pyigqn/_blank)和其他语言编写的项目。Maven曾是[Jakarta项目](https://zh.wikipedia.org/wiki/Jakarta%E9%A1%B9%E7%9B%AE" \t "https://www.yuque.com/cpwen/pyigqn/_blank)的子项目，现为由[Apache软件基金会](https://zh.wikipedia.org/wiki/Apache%E8%BD%AF%E4%BB%B6%E5%9F%BA%E9%87%91%E4%BC%9A" \t "https://www.yuque.com/cpwen/pyigqn/_blank)主持的独立Apache项目。它有许多优点，例如：

**管理模块之间的依赖**：根据业务需求，系统会划分很多模块，这些模块彼此之间存在着依赖关系。比如财务模块依赖着用户模块，来实现读取用户信息的功能。maven通过配置模块之间的pom文件进行依赖。

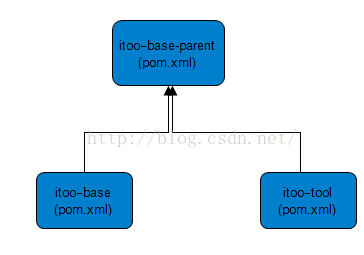


图2-1-2 pom继承关系图

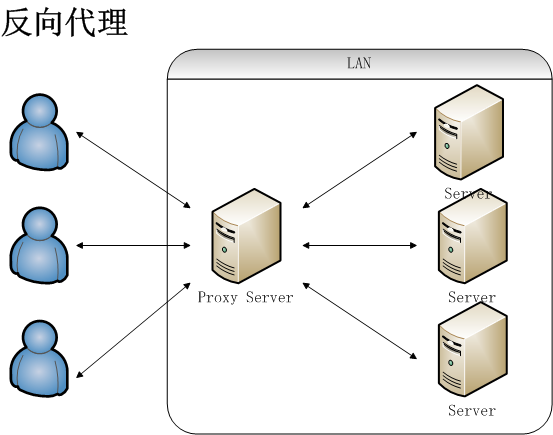
**生命周期管理**：在web应用中，我们常要进行编译、打包、测试这些环节。在maven的世界里，将这些过程定义为生命周期。Maven为开发人员构建了一个完整的生命周期框架。开发团队可以自动完成项目的基础工具建设，Maven使用标准的目录结构和默认构建生命周期。在多个开发团队环境时，Maven可以设置按标准在非常短的时间里完成配置工作。由于大部分项目的设置都很简单，并且可重复使用，Maven让开发人员的工作更轻松，同时能创建报表，检查，构建和测试自动化设置。

**强大的插件**：maven中有许多功能强大且使用方便的插件。举一个很常用的插件tomcat7-maven-plugin，在以前，我们发布应用的方式是，将web应用打包成war->手动拷贝到tomcat的webapp目录下->启动tomcat。现在有了Maven，以上过程我们只需要点击一下鼠标即可完成。

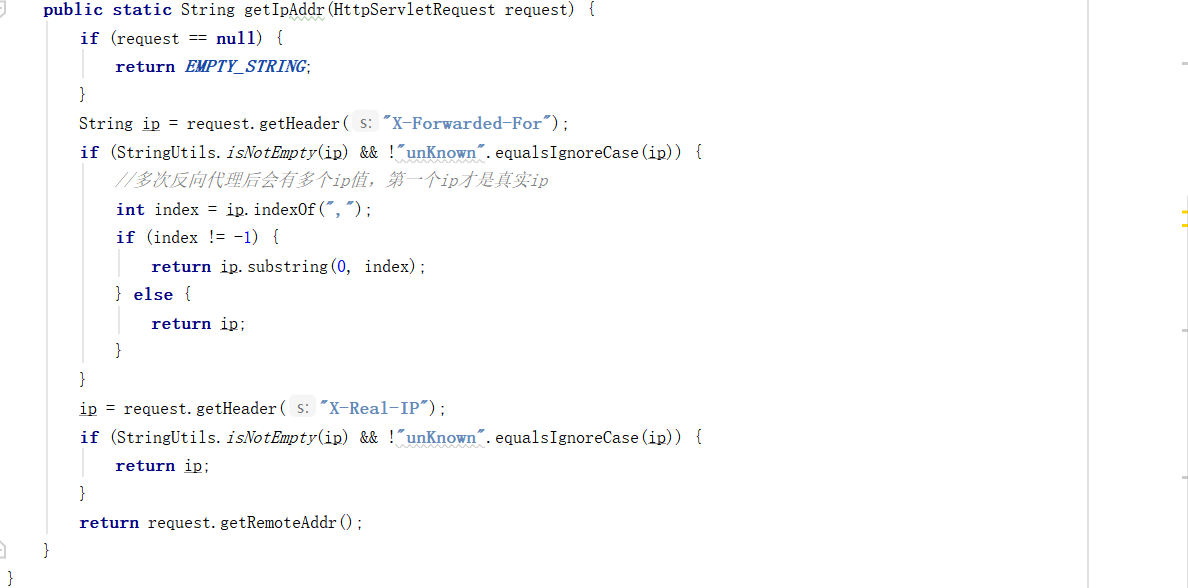
**仓库式管理**：Maven提供公用仓库http://mvnrepository.com/，只要输入jar包的坐标即可查找到想要的资源，将模块的坐标加入到自己的pom中就可以快乐的使用了，同时由于依赖的概念，子模块能够继承父模块的版本号，这样就可以统一管理项目的版本号，确保应用的各个项目的依赖和版本一致。

**2.1.3 Nigix反向代理**

反向代理（Reverse Proxy）方式是指以代理服务器来接受Internet上的连接请求，然后将请求转发给内部网络上的服务器，并将从服务器上得到的结果返回给Internet上请求连接的客户端，此时代理服务器对外就表现为一个服务器。



第二课堂小程序面向校园广大学生,考虑到大型活动,众多学生同时访问业务需求.由于单个服务器处理微信端请求能力有限,当学生的请求访问数量巨大,会造成服务器忙不过来的局面,这时可使用多个服务器来一起分担当前学生请求.在nigix中配置真实服务器地址,接收客户端请求,后将请求分发给不同服务器进行处理,然后在将服务器的响应结果反馈给客户端,以此来避免请求量过大情况.



**2.1.4 代码规范**

1. 遵循restful接口规范

一种软件架构风格、设计风格，而**不是**标准，只是提供了一组设计原则和约束条件。它主要用于客户端和服务器交互类的软件。基于这个风格设计的软件可以更简洁，更有层次，更易于实现缓存等机制。

1. 遵循阿里巴巴开发规范

来自于阿里集团的智慧与经验，用血淋淋的故障换来的总结 阿里经济体全球技术开发准则

定制团队成员内部遵守规范

1. 所有class,interface,enum需详细注解
2. **所有Interface的接口上面需有接口描述**
3. 所有成员变量上需要注解描述含义
4. 复杂的boolean判断需拆出，严禁出现 switch等没有明确寓意 的boolean变量，建议使用switchOn/switchOff
5. 代码中不能出现魔法值，尽量使用常量、常量类或枚举类管理
6. 注意PMD问题(不必要的变量/引用等)
7. 数据库建库时索引命名规范, 唯一索引名称用 uk\_xxx , 一般索引名称 idx\_xxx。 xxx为对应的字段名称  
   例如: idx\_open\_id, uk\_user\_id\_user\_info\_id等

### 2.2前端技术介绍:

### 2.2.1小程序技术介绍:

1. **主要开发模式：**

在开发本项目的移动端应用程序之前，我们考虑了它能够运行的平台，比如Android或iOS。应用可以大致分为Web和移动应用，移动应用程序可以进一步划分为Native App和Hybrid App。

**Native App:**



选择Native app的优势在于它在用户体验方面是最快和最可靠的，可脱机工作，可以访问所有设备级功能。Native app针对特定平台（如iOS，Android和Window手机）创建，因此如果想要创建跨平台应用，就需要单独针对多平台进行开发，维护和更新成本较高。

**Web App：**

****

Web app(H5)开发是提供移动用户体验的成本最低的选择，Web app使用HTML，CSS和JavaScript构建。加载在Chrome或Firefox等浏览器上，不占用用户设备上的内存或存储空间，功能齐全，无需像移动应用程序一样下载。Web app易于维护，开发成本较低。

然而缺点也是显而易见的，与其他应用相比，交互不是那么整洁和敏感，页面切换时会出现白屏，影响用户体验。对Internet连接依赖很高。

**Hybrid App：**

****

在某种程度上，Hybrid App是Web和Native开发之间的折衷。它结合了两种类型的app开发的优点。

Hybrid app在技术上是一个打包在Native容器中的Web混合类App。与Web App一样，它是用HTML，CSS和JavaScript编写的。不同的是，它通过应用商店进行发布，可脱机工作，简单快速的更新。

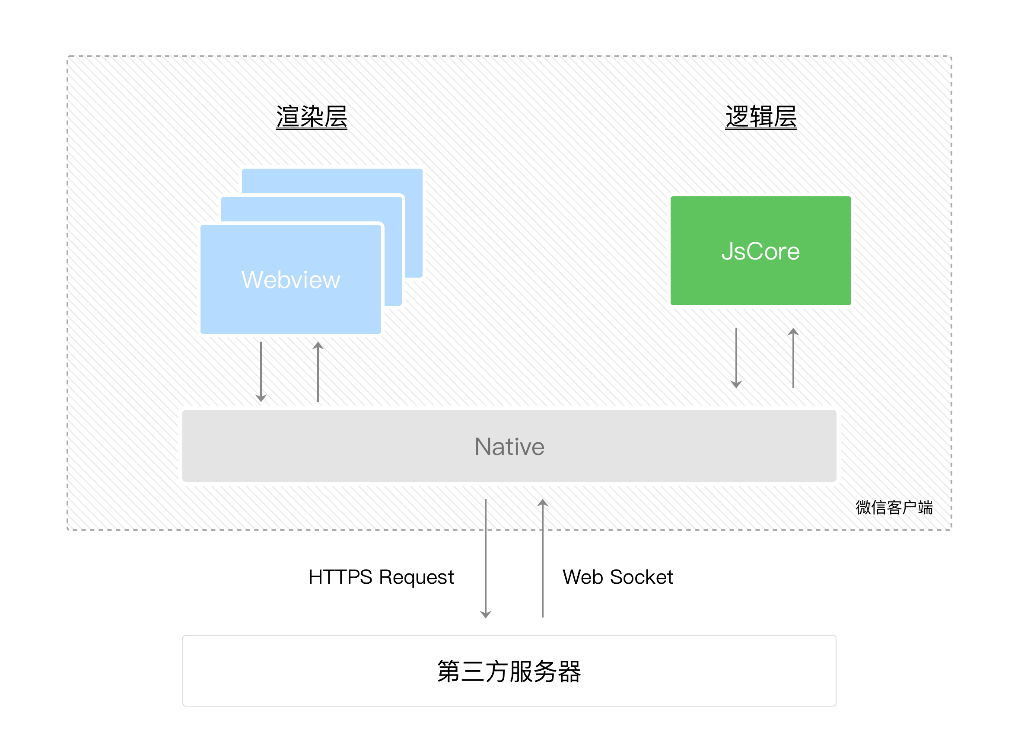
**设计要点：**



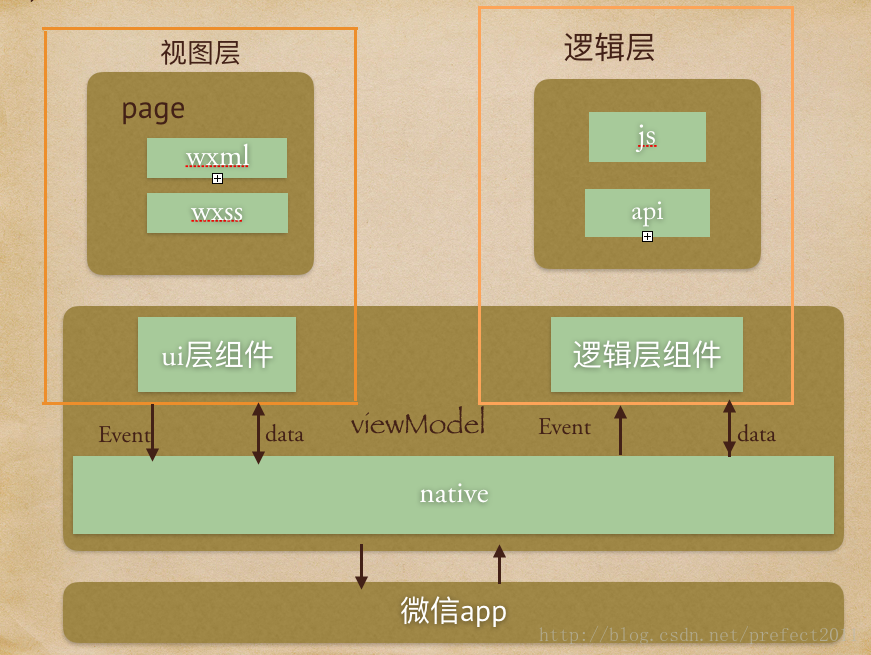
微信小程序采用了Hybrid app模式，由上图可见三种开发模式的优劣势，鉴于此，选择了微信小程序。小程序即点即用，用户使用成本低。在速度方面，主要样式代码都封装在微信小程序里，打开速度比H5快，慢于原生。同时，小程序可以调用更多的手机硬件功能，例如GPS，录音。由于支持微信的在线版本更新，开发、维护成本较低，同时维护几个版本的可能性较小。

**技术特性：**

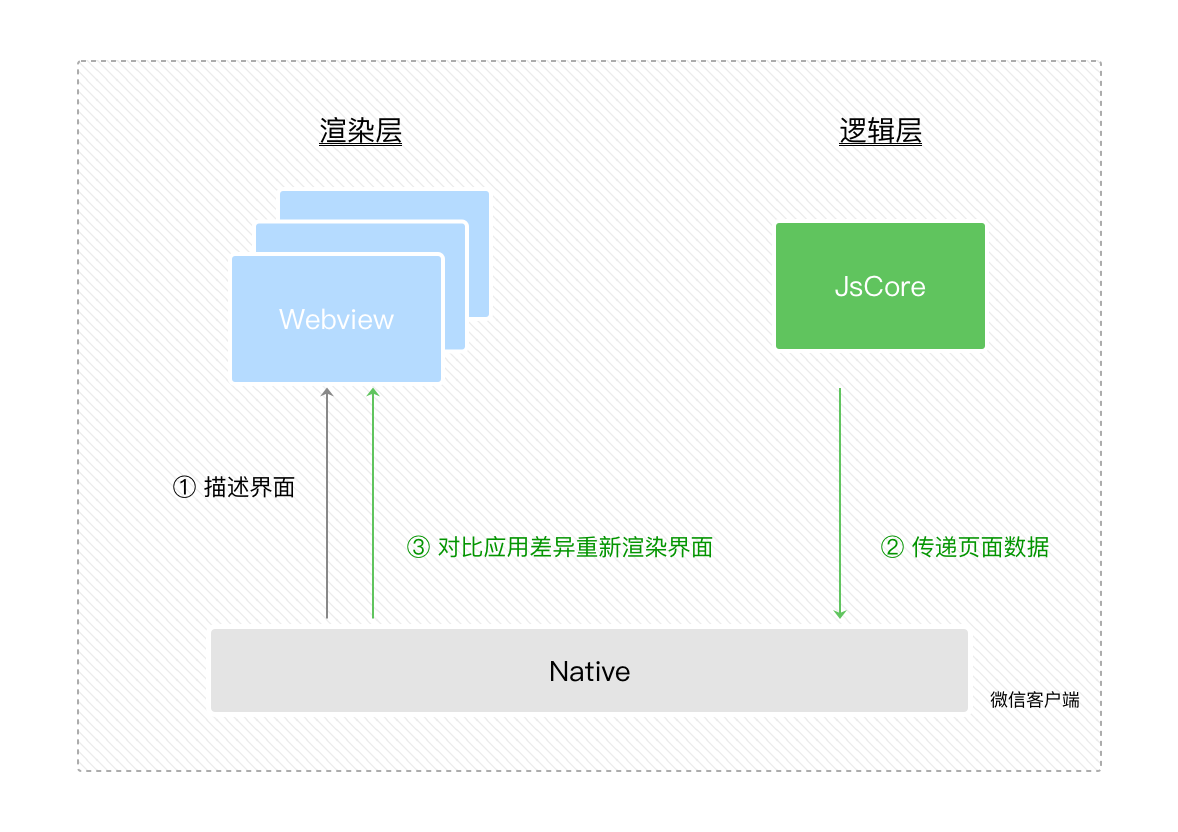
小程序采用了 Hybrid 的渲染方式，结合了Web 与 Native 两者的优点，以Web的方式来开发，同时借助微信平台，支持版本的在线更新。为了获得更好的体验，小程序将渲染层和逻辑层分离，使渲染的性能更好，减轻了WebView的渲染压力，这一思想，小程序采用双线程模型实现：



将渲染层和逻辑层分离。逻辑层负责业务逻辑的处理，由单独的一个线程处理，执行JavaScript。渲染层负责界面的渲染，由逻辑层控制渲染的部分，渲染层存在多个 WebView 线程。



逻辑层和渲染层的通信会经由微信客户端（Native）做中转，逻辑层发送网络请求也经由 Native 转发。小程序无法直接操作DOM，采用了数据绑定和虚拟DOM的方式实现页面渲染。



1. **前端开发语言：**

**WXML**

WXML是微信小程序框架设计的一套类似HTML的标签语言，在HTML语言上做了一些改变，采用小程序原生的组件标签，结合基础组件、事件系统，可以构建出页面的结构，熟悉网页开发即可快速掌握。WXML具有数据绑定、列表渲染、条件渲染、模板及事件绑定的能力。仅能在微信小程序开发工具中预览，WXML对组件进行了重新封装，为后续的性能优化提供了可能，同时避免开发者写出低质量的代码。

**WXSS**

WXSS用于描述WXML组件的样式，其具有CSS的大部分特性，同时为了更适合微信小程序的开发，对CSS进行了扩充和修改。相比于CSS，扩展的特性有：尺寸单位和样式导入。

尺寸单位引入了rpx，可根据手机屏幕的宽度进行自适应。



样式导入使用@import语句可以导入外联样式表，@import后跟需要导入的外联样式表的相对路径

**JavaScript**



JavaScript对WXML页面进行响应交互，小程序中的JavaScript是由ECMAScript以及小程序框架和小程序 API 来实现的。由于渲染层和逻辑层分离，没有DOM对象，因此无法使用JQuery、Zepto等的类库。

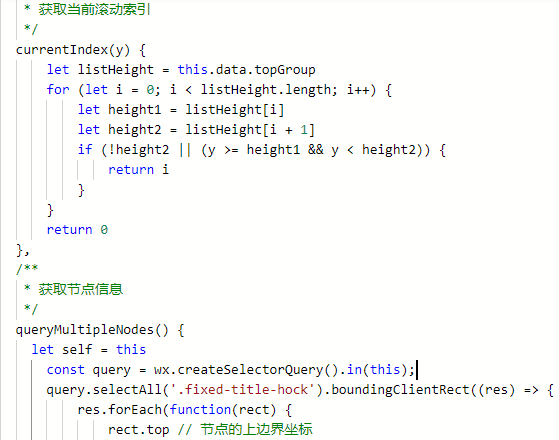
1. **小程序登录授权**

通过微信官方提供的wx.login()函数获取临时登录凭证code，并回传到开发者服务器，开发者服务器通过登录凭证校验接口传输appid+appsecret+code到微信接口服务，返回获得session\_key+openid，开发者服务器根据openid向小程序前端返回token，作为用户唯一标识符，小程序获取后存在本地，每次由wx.request()发起请求时，HTTP请求头Authorization携带token验证用户权限信息。具体流程如下：



**UI组件库**

为增强用户体验，本项目采用了Wux WeApp的UI组件库，统一页面的风格。同时根据需求，部分模块重合度较高，单独的组件会产生不必要的冗余，因此组件的复用可很大程度上提高项目的可维护性。我们自行封装了一些自定义组件，如列表索引组件，功能选项组件。



为了提高用户交互体验，我们为小程序组件添加了动画过渡效果。由于小程序不可直接操作DOM，自带的API动画接口wx.createAnimation效率较低，考虑到WXSS支持CSS3的特性，因此采用了CSS3实现动画效果，不占用JS主线程，解决了动画带来的性能问题。

1. **学生端二维码**

第二课堂小程序学生端的个人信息二维码由小程序本地生成，二维码携带信息为json格式，具体内容为：

{

"legal":"No2Class",

"stuId":"17905215",

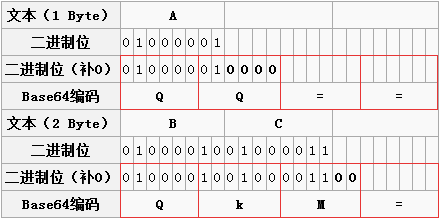
"stuName":"XXX",

"timestamp":1554900949268

}

为保证信息的安全性，二维码信息采用BASE64算法进行加密，防止肉眼可以直视出个人信息的处理规则。

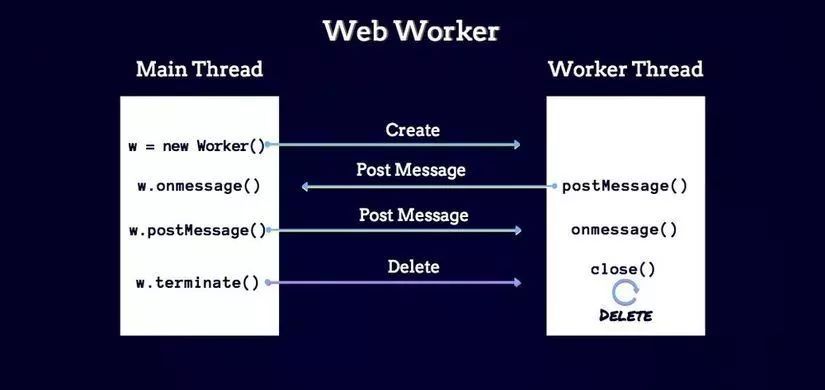
Base64是网络上最常见的用于传输8Bit字节码的编码方式之一，索引表字符选用了"A-Z、a-z、0-9、+、/" 64个可打印字符，Base64就是一种基于64个可打印字符来表示二进制数据的方法。Base64的码表只有64个字符， 如果要表达64个字符的话，使用6的bit即可完全表示(2的6次方为64)。因为Base64的编码只有6个bit即可表示，而正常的字符是使用8个bit表示， 8和6的最小公倍数是24，所以4个Base64字符可以表示3个标准的ASCII字符。



小程序在扫码员权限下获取二维码解析，验证信息的正确性，进行BASE64解码，即可获取原始数据，信息对称安全。

此外，为了保证二维码的时效性，在二维码信息中加入了时间戳，并10秒刷新一次，可防止活动章代签的情况发生，同时在扫码员端设置20秒的时间冗余，减少因手机延迟产生的误差。

在生成二维码的同时js后台计时，为防止线程阻塞，采用了js的多线程处理。因为Js本身是单线程，无法处理多线程，虽然通过类似定时器、回调函数等异步编程方式，可模拟实现多线程处理，但很容易触发假死状态，异步代码会影响主线程的代码执行，异步终究还是单线程，不能从根本上解决问题。采用Woker多线程很好的解决了这一问题，在主线程运行的同时，Worker（子）线程在后台运行，两者互不干扰。等到 Worker 线程完成计算任务，再把结果返回给主线程。

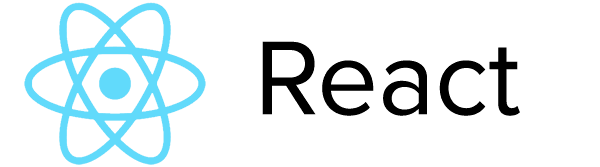


### 2.4网页技术介绍:

为了能够更直观的展示后台数据，进行数据的批量处理，提高效率，我们开发了后台管理系统，对各个系统模块进行统一管理。

**技术实现：**

**React**



后台管理系统采用了React框架，提高开发效率。在Web开发中，我们总需要将变化的数据实时反应到UI上，这时就需要对DOM进行操作。而复杂或频繁的DOM操作通常是性能瓶颈产生的原因。React为此引入了虚拟DOM（Virtual DOM）的机制，每当数据变化时，React都会重新构建整个DOM树，然后React将当前整个DOM树和上一次的DOM树进行对比，然后仅仅将需要变化的部分进行实际的浏览器DOM更新。不同于Jquery等框架的直接操作DOM，性能有了很大提升。

虚拟DOM(virtual-dom)不仅带来了简单的UI开发逻辑，同时也带来了组件化开发的思想， UI功能模块之间相互分离，每个组件都只关注自己部分的逻辑，彼此独立。

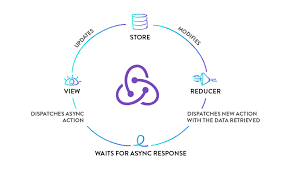
在本项目的构建中，考虑到系统的实时性能，以及实时的数据监测，部分模块具有一定的复用性，因此选择了React进行开发。

**Redux**



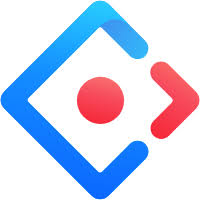
React中的状态管理十分的庞大，react中的状态可能来自服务器、缓存数据、UI状态，或者激活路由等，管理不断变化的状态是十分困难的，这个时候，我们引用redux来对react中的状态进行管理，让状态的管理更加的简洁和方便。

**Redux-thunk**



Redux可以很方便的利用Action，Reducer函数更新state来重新渲染View。但是其中的异步问题没有办法解决。这个时候我们就利用了redux的中间件--redux-thunk来解决这个问题，redux-thunk通过使得action操作脱离正常的流程，来到中间件进行执行，让页面的异步请求的问题得到解决。

**Antd**



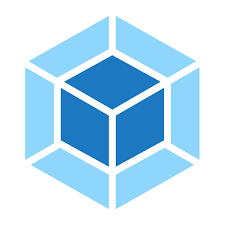
后端页面的实现功能的同时也要注重美化和方便，为此antd-react成为了不二之选，antd提供了大量丰富的组件，使得页面的搭建更加便捷，增加了页面的美观度、加快了页面框架的搭建以。

**Axios**



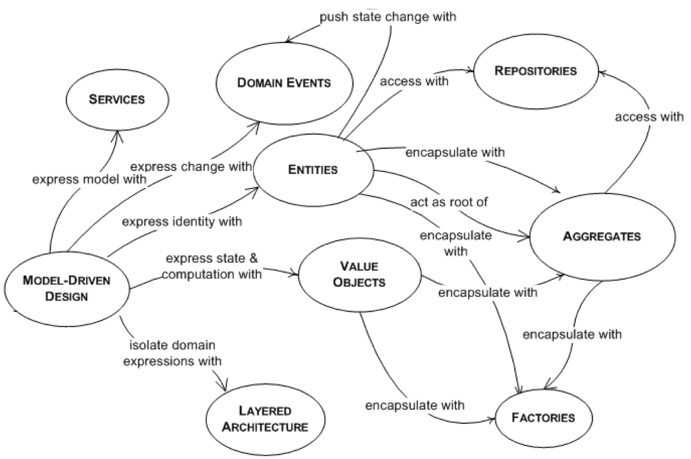
Axios是一个基于promise的HTTP库，可以用在浏览器和node.js。Axios的现在流行的网络库之一。在项目中，我们将axios进行二次封装，使得axios的功能更加强大，也让我们的网络请求变得更加的简便。以及解决前后端分离中的常见问题--跨域

**Webpack**



Webpack作为现在最流行的项目打包程序，自然是我们项目架构中的最优选择。我们利用webpack自定义了开发框架和流程，不再依赖react的create-react-app命令，实行了按需添加。并且使用热加载技术，使得开发效率得到极大的提升

### 2.5领域驱动设计思想

****

**领域模型**（或称**域模型**；英语：domain model）可以被看作是一个系统的概念[模型](https://zh.wikipedia.org/wiki/%E6%A8%A1%E5%9E%8B" \t "https://www.yuque.com/cpwen/pyigqn/_blank)，用于以可视化的形式描述系统中的各个实体及其之间的关系。领域模型记录了一个系统中的关键概念和词汇表，显示出了系统中的主要实体之间的关系，并确定了它们的重要的方法和属性。因此，对应于[用例](https://zh.wikipedia.org/wiki/%E7%94%A8%E4%BE%8B" \t "https://www.yuque.com/cpwen/pyigqn/_blank)所描述的动态视图，领域模型提供了一种对整个系统的结构化的视图。领域模型的一个好处是描述并限制了系统边界。

领域模型的语义可以被用在源代码中，因此领域模型可以被应用在底层的软件开发阶段中。实体可以演化为类，方法和属性可以直接演化至代码之中。

**第三部分 第二课堂框架&架构设计**

**3.1 后端框架**

**3.1.1 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。

**3.1.2 Spring**

Spring是一个开源框架，Spring是于2003 年兴起的一个轻量级的Java 开发框架，由Rod Johnson在其著作Expert One-On-One J2EE Development and Design中阐述的部分理念和原型衍生而来。它是为了解决企业应用开发的复杂性而创建的。Spring使用基本的JavaBean来完成以前只可能由EJB完成的事情。然而，Spring的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何Java应用都可以从Spring中受益。简单来说，Spring是一个轻量级的控制反转（IoC）和面向切面（AOP）的容器框架。

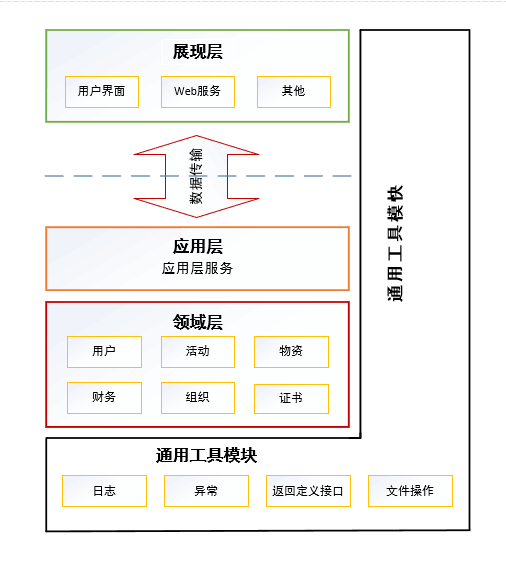
**3.1.3 Spring date jpa**

JPA(Java Persistence API)是Sun官方提出的Java持久化规范。它为Java开发人员提供了一种对象/关联映射工具来管理Java应用中的关系数据。他的出现主要是为了简化现有的持久化开发工作和整合ORM技术，结束现在Hibernate，TopLink，JDO等ORM框架各自为营的局面。值得注意的是，JPA是在充分吸收了现有Hibernate，TopLink，JDO等ORM框架的基础上发展而来的，具有易于使用，伸缩性强等优点。Spring Data JPA 是 Spring 基于 ORM 框架、JPA 规范的基础上封装的一套JPA应用框架，可使开发者用极简的代码即可实现对数据的访问和操作。它提供了包括增删改查等在内的常用功能，且易于扩展！学习并使用 Spring Data JPA 可以极大提高开发效率！spring data jpa让我们解脱了DAO层的操作，基本上所有CRUD都可以依赖于它来实现.

**3.2架构设计**

**3.2.1总体介绍**

项目初期.第二课堂小程序功能较为单一,普通的CRUD就能满足需求.故在整体项目使用MVC模式开发,系统结构简单清晰.随着项目功能不断增加,业务逻辑变得越来越复杂,第二课堂变得越来越冗杂,对于项目不利于维护.模块之间耦合度高,不符合软件设计规范中低耦合高内聚的思想,故进行项目重构使用DDD领域模型设计.



第二课堂使用分层架构中的四层架构:

1. |-- biz （业务层）

|-- impl (业务逻辑实现)

1. |-- core (独立的业务领域核心)

|-- user (用户领域核心)

|-- activity (活动领域核心)

|-- asset (物资领域核心)

|-- finance (财务领域核心)

|-- organization(组织领域核心)

|-- certificate (证书领域核心)

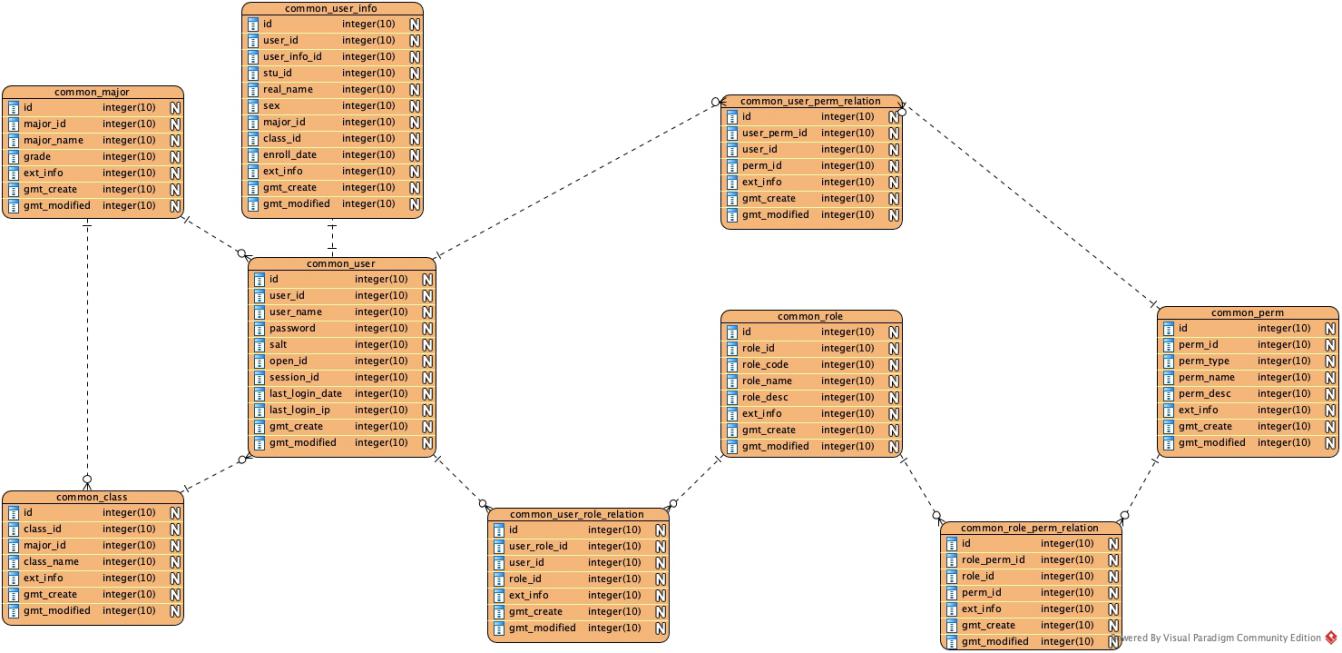
1. |-- web (接口交互层)
2. |-- util (通用工具模块)

**3.2.2领域层**

领域层(core层)以用户领域、活动领域、物资领域、财务领域、组织领域、证书领域为核心构成核心模块,biz层依赖core层.实现领域交际功能。

1. 用户领域

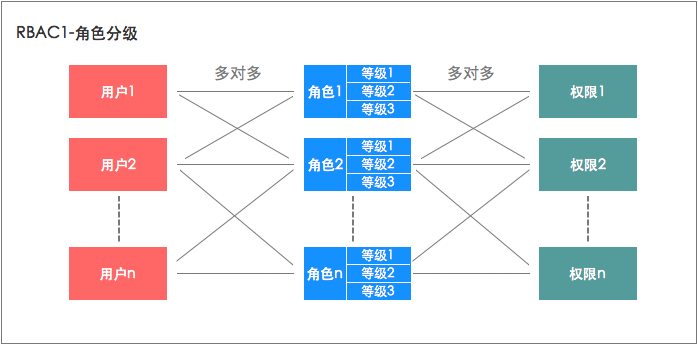
用户领域是第二课堂第一个搭建模块,小程序用户以学生为主体,面向学生属性进行数据库设计。



**Rbac模型**

RBAC是一套成熟的权限模型。在传统权限模型中，我们直接把权限赋予用户。而在RBAC中，增加了“角色”的概念，我们首先把权限赋予角色，再把角色赋予用户。这样，由于增加了角色，授权会更加灵活方便。在RBAC中，根据权限的复杂程度，又可分为RBAC0、RBAC1、RBAC2、RBAC3。其中，RBAC0是基础，RBAC1、RBAC2、RBAC3

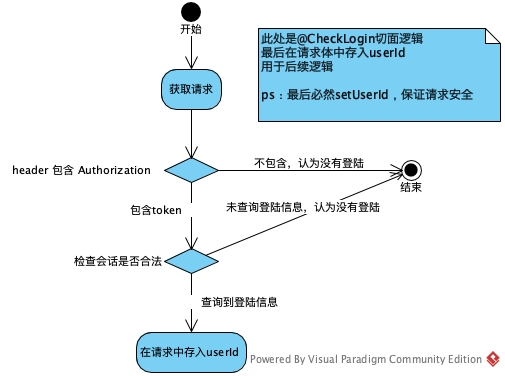
都是以RBAC0为基础的升级。对此第二课堂以RBAC1为蓝本建立完善的权限管理模型以保证最小权限原则，责任分离原则和数据抽象原则。



**Jwt请求认证**

Json web token (JWT), 是为了在网络应用环境间传递声明而执行的一种基于JSON的开放标准（[(RFC 7519](https://link.jianshu.com/?t=https://tools.ietf.org/html/rfc7519" \t "https://www.yuque.com/cpwen/pyigqn/_blank)).该token被设计为紧凑且安全的，特别适用于分布式站点的单点登录（SSO）场景。JWT的声明一般被用来在身份提供者和服务提供者间传递被认证的用户身份信息，以便于从资源服务器获取资源，也可以增加一些额外的其它业务逻辑所必须的声明信息，该token也可直接被用于认证，也可被加密。

传统身份验证使用HTTP协议,用户发送请求到服务器,服务器创建唯一session记录存储到Cookie中.用户再次提交请求时,服务器对Cookie中session进行身份验证.使用session存储方式相对cookie来说更安全,但由于"第二课堂"使用nigix利用反向代理来实现负载均衡,导致下一个操作请求提交到另外一台服务器的时候session会丢失.故为解决该难点,系统使用JWT认证,服务端无需保存用户的认证信息或者会话信息,在提交登陆或者操作请求时,将token信息放入Header请求头Authorization中并使用Https对请求链接进行加密传输,后端对token参数进行验证后提供后续服务.最后会话合法,在请求中获取userId,保证请求安全.



**Validations数据检验器**

Hibernate中不仅有ORM一个框架，它的另一个框架Validator也是被经常使用，用来验证实体类是否满足需求。Validator实现了Java的一项标准Bean Validation。

"第二课堂"小程序,不仅实现数据导入,根据学生名单自动创建学号账号,并且用户可以创建账号。

数据校验是用户注册部分不可或缺的功能,小程序前端可对数据进行第一层校验,如账号密码合理性,但是为了避免用户绕过浏览器,使用postman等工具向后端传入违法数据,服务端的数据校验也是必须的.对此引入hibernate validation (JSR303).

后端继承validation实现openid校验器,密码长度校验器,密码强度校验器,用户账号校验器.

密码强度校验器:对密码进行等级计算,当密码含有数字时密码强度+1,对密码中是否含有大小写字符进行判断,密码长度分别+1,对特殊符号判断,密码强度+1,密码强度逐层累加,当密码强度小于2时,强度校验器返回异常信息,用户创建账号失败.

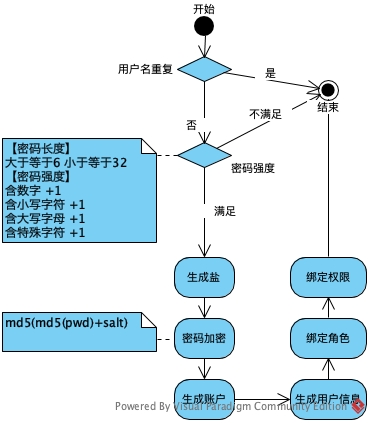
用户账号校验器:当用户提交创建账号请求,获取提交账号参数中用户账号参数,数据库验证是否有相同账号若有则返回异常信息,用户创建账号失败.

**MD5加盐算法**

**MD5**即Message-Digest Algorithm 5（信息-摘要算法5），用于确保信息传输完整一致。是计算机广泛使用的杂凑算法之一（又译[摘要算法](https://baike.baidu.com/item/%E6%91%98%E8%A6%81%E7%AE%97%E6%B3%95/12011257" \t "https://www.yuque.com/cpwen/pyigqn/_blank)、[哈希算法](https://baike.baidu.com/item/%E5%93%88%E5%B8%8C%E7%AE%97%E6%B3%95/4960188" \t "https://www.yuque.com/cpwen/pyigqn/_blank)），主流编程语言普遍已有MD5实现。将数据（如汉字）运算为另一固定长度值，是杂凑算法的基础原理，MD5的前身有MD2、[MD3](https://baike.baidu.com/item/MD3" \t "https://www.yuque.com/cpwen/pyigqn/_blank)和[MD4](https://baike.baidu.com/item/MD4" \t "https://www.yuque.com/cpwen/pyigqn/_blank)。

现在的MD5密码数据库的数据量已经非常庞大了，大部分常用密码都可以通过MD5摘要反向查询到密码明文。为了防止内部人员（能够接触到数据库或者数据库备份文件的人员）和外部入侵者通过MD5反查密码明文，更好地保护用户的密码和个人帐户安全（一个用户可能会在多个系统中使用同样的密码，因此涉及到用户在其他网站和系统中的数据安全），第二课堂小程序采用MD5加盐算法,即对MD5摘要结果掺入其他信息，称之为加盐。

MD5加盐算法是针对每一个用户U，生成一个随机值Salt，并且在以后永远保持不变，任意两个用户的盐不能相同。然后当用户设置密码的时候，根据明文密码P，计算MD5(P+Salt)=C。而登录的时候用户也给出明文密码P'，服务器拿到之后同样计算MD5(P'+Salt)=C'，看C'是否等于C。故使用MD5加密算法,极大增加黑客破解难度和时间.更好的保护了用户的密码和账号安全.



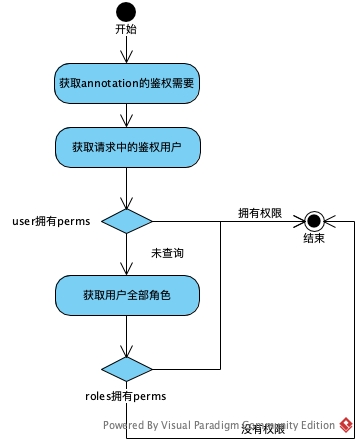
**权限注解+AOP鉴权**

Annotation（注解）就是Java提供了一种元程序中的元素关联任何信息和着任何元数据（metadata）的途径和方法。Annotion(注解)是一个接口，程序可以通过反射来获取指定程序元素的Annotion对象，然后通过Annotion对象来获取注解里面的元数据。

 Annotation(注解)是JDK5.0及以后版本引入的。它可以用于创建文档，跟踪代码中的依赖性，甚至执行基本编译时检查。从某些方面看，annotation就像修饰符一样被使用，并应用于包、类 型、构造方法、方法、成员变量、参数、本地变量的声明中。这些信息被存储在Annotation的“name=value”结构对中。

AOP，即面向切面编程。AOP能够将那些与业务无关，却为业务模块所共同调用的逻辑或责任，例如事务处理、日志管理、权限控制、异常处理等，封装起来，便于减少系统重复的代码，降低模块之间的耦合度。

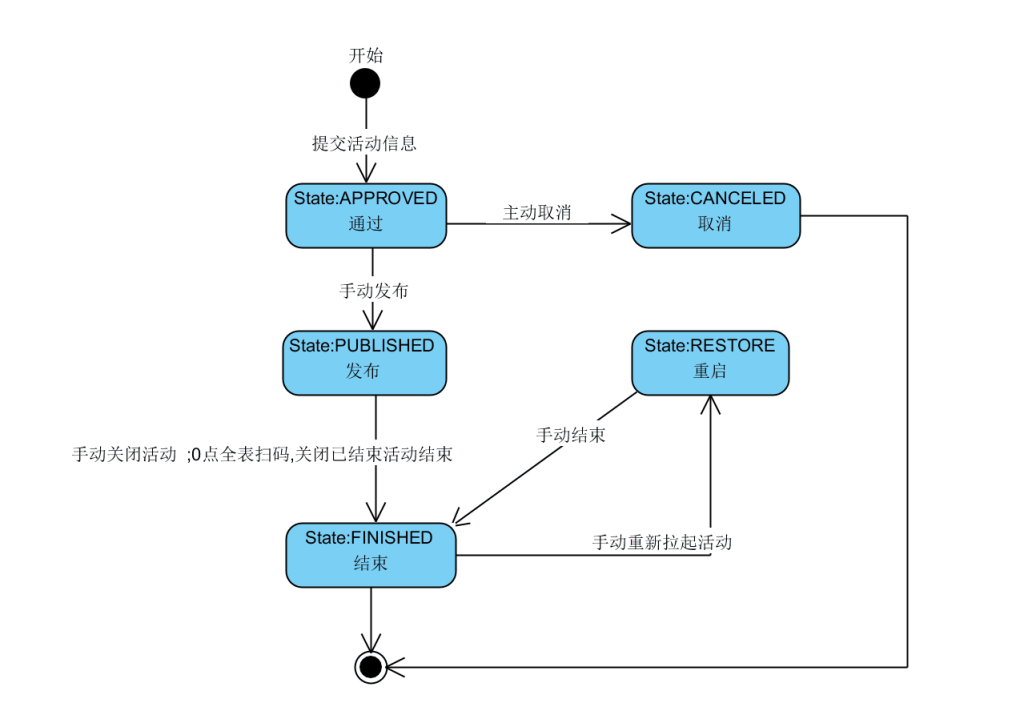
传统鉴权方法使用springmvc方式对用户身份进行拦截,"第二课堂"鉴权采用注解配合AOP切面实现拦截,后端通过自定义@VerifyPerm注解对登陆用户拥有权限查询获取所有权限返回前端,前端对用户权限进行不同页面跳转.



1. 活动领域:

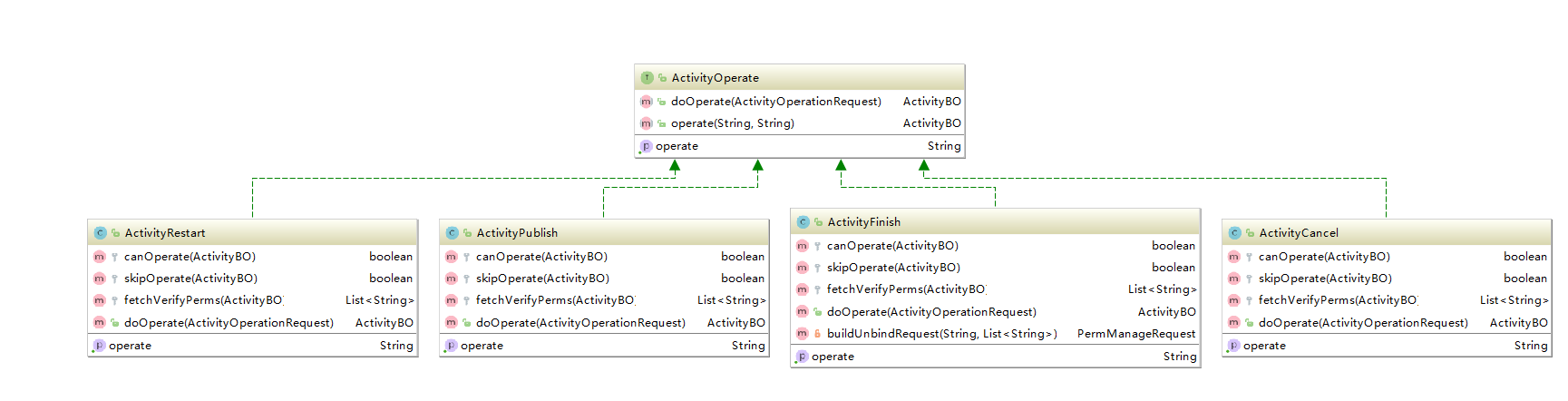
传统纸质材料管理学生活动信息，不仅繁琐耗时，而且可能会出现管理漏洞和僵化管理的问题。现在通过技术手 段管理学生校园活动、志愿活动、义工、社会实践等信息，希望能改善现状，优化管理和学生的体验。

**a 状态机模式**



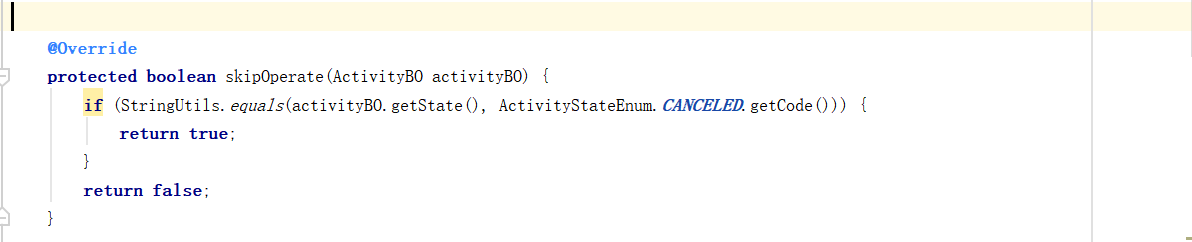
活动状态转换图

当活动操作员对活动状态进行修改时,后端需要对当前活动状态与操作活动转换状态进行合理化判断,业务代码充斥大量条件判断,各种状态处理逻辑散列各处,不利于代码理解与修改,新增活动需要添加多重判断,不利于功能添加与测试违反了软件设计中"开放与封闭原则".为此,我们将状态模式的概念付诸实施，将与指定状态有关的逻辑操作分别写在对应的可代表状态的类里。



**b 幂等操作**

**幂等**（idempotent、idempotence）是一个数学与计算机学概念，常见于抽象代数中。 在编程中一个**幂等操作**的特点是其任意多次执行所产生的影响均与一次执行的影响相同。 **幂等**函数，或**幂等**方法，是指可以使用相同参数重复执行，并能获得相同结果的函数。

为避免活动管理员对一个活动的状态进行多次重复提交,数据库数据多条重复数据,后端对活动状态更新接口实现幂等操作.由此相同的参数调用一次接口和调用多次其返回结果值相同.  


1. 组织领域

组织系统分析

**总体概括**

为记录学校各大组织成员任职更替频繁记录,弥补纸质证书盖章效率低下且证书丢失补盖证书周期长等问题.我们新增了组织领域.

组织领域 :

数据库记录表为三张:组织表:记录学校目前登记注册组织

组织关系表:组织之间从属关系

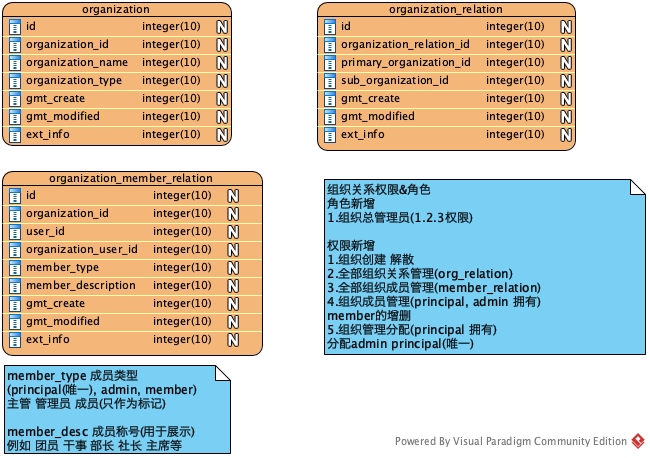
组织成员关系表:记录学生归属组织及职务

组织关系权限和角色

角色包括:组织总管理员、各组织管理员、成员

组织总管理员:拥有组织创建 解散,全部组织关系管理,全部组织成员管理权限

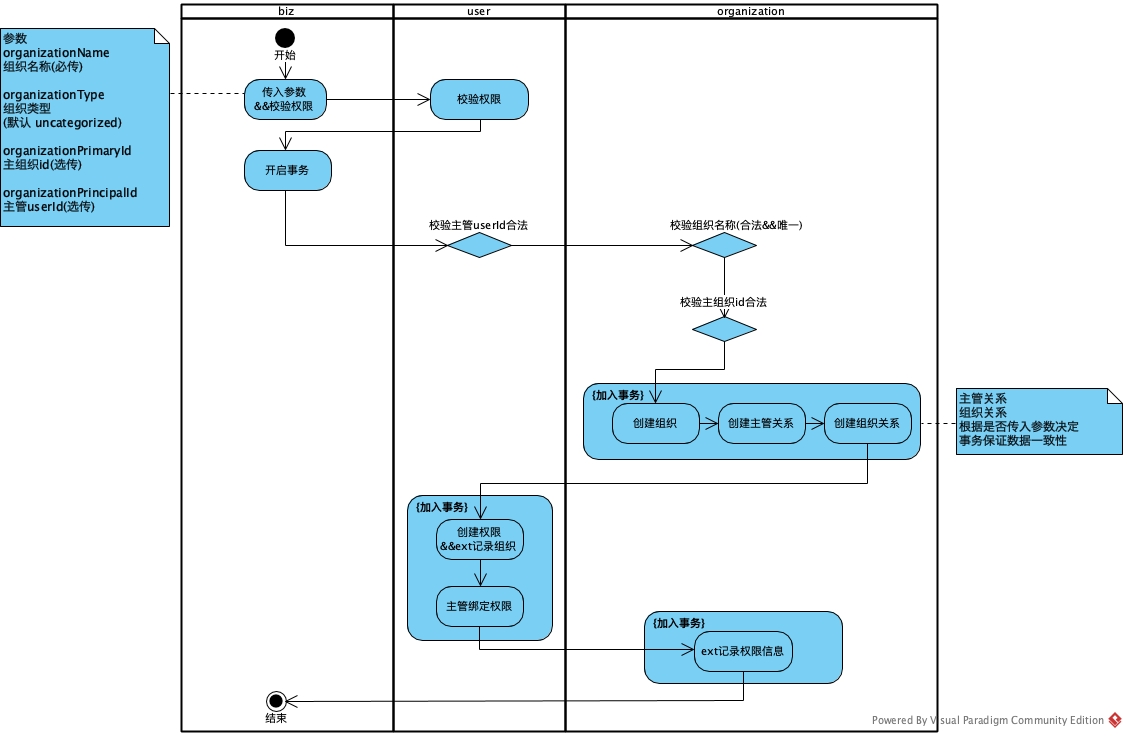
当学生向学校申请创建组织,审核通过,组织总管理员对组织进行创建.当某组织会长(最该职位进行替换时)对原该组织管理员权限删除.避免出现同一组织多管理员使组织内部管理混乱.组织管理员可对该组织成员进行增删.成员称号进行修改。



**创建组织**

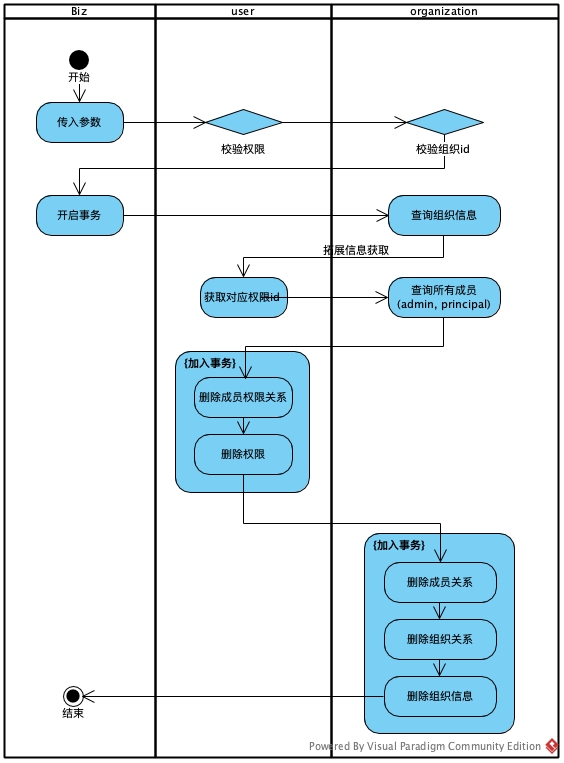
组织总管理员根据通过审核数据创建组织,通过POST请求带上四个参数(organizationName,organizationType,organizationprimaryid,organizaitonPrincipalld).后端通过@checkLogin自定义注解,获取登录用户权限,判断此条请求合法性.若合理则对组织管理员学号

组织名称.组织归属组织id进行合法性判断.信息完整无误后使用spring事务注解@Transaction开启事务,创建从属组织之间绑定关系.创建组织管理员权限而非组织管理员避免角色复杂化,将创建权限id放入该组织额外信息中并与传入用户学号进行绑定。



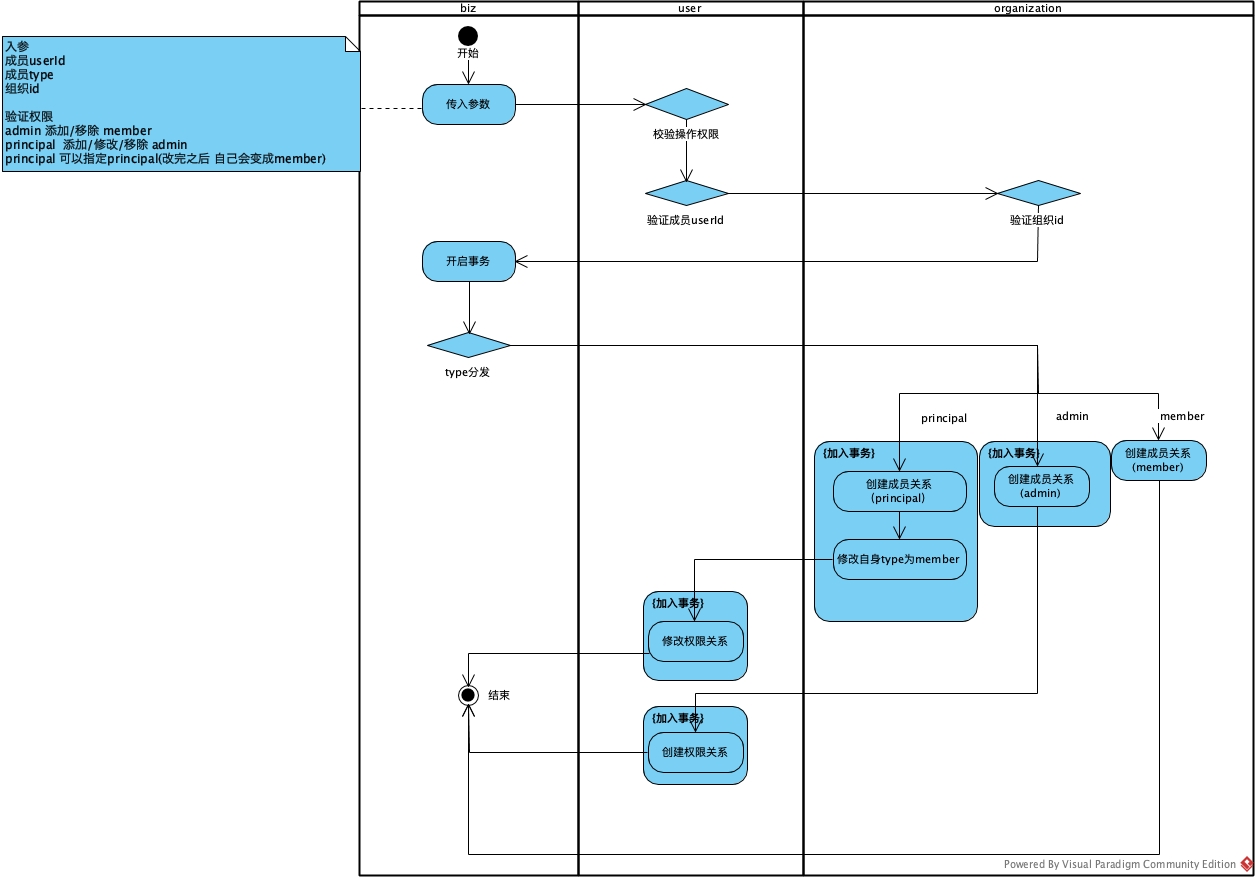
**组织解散**

组织解散是组织模块必不可少的一环,第二课堂考虑长期项目使用,部分组织应各种原因,组织无法继续带领学生成长,可由总管理员对组织进行删除.系统内部实现对该组织成员与权限统一删除.操作简洁明了。



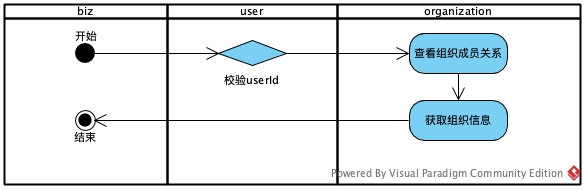
**成员管理**

成员管理是组织模块必不可缺的部分,不同权限用户通过权限类型进行分发管理,管理员通过成员管理,使各成员可清晰查看自己社团职位.同时社团主席选举换届,社团管理员采用转让制,而无须总管理员进行后台修改.减少行政审核时间,并在任一时间始终保持社团管理员单一制。



**查看组织**

对用户权限判断,为总管理员或管理员才能获取组织成员关系



**3.2.3应用层**

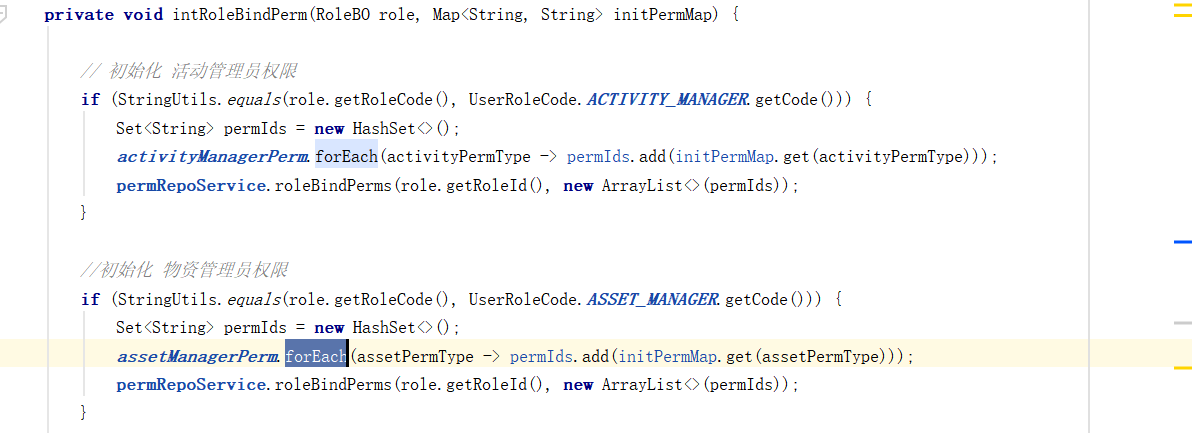
Bizbiz 层依赖所有core模块，处理领域交际部分逻辑

**Stream流操作**

Stream API是Java 8中加入的一套新的API，主要用于处理集合操作，不过它的处理方式与传统的方式不同，称为“数据流处理”。

Stream接口中包含许多对流操作的方法，常用方法有：

1. filter()：对流的元素过滤map()：将流的元素映射成另一个类型
2. distinct()：去除流中重复的元素
3. forEach()：对流中的每个元素执行某个操作
4. toArray()：将流转换为数组
5. collect()：对流的汇总操作，比如输出成List集合



**函数式编程--- Lambda**

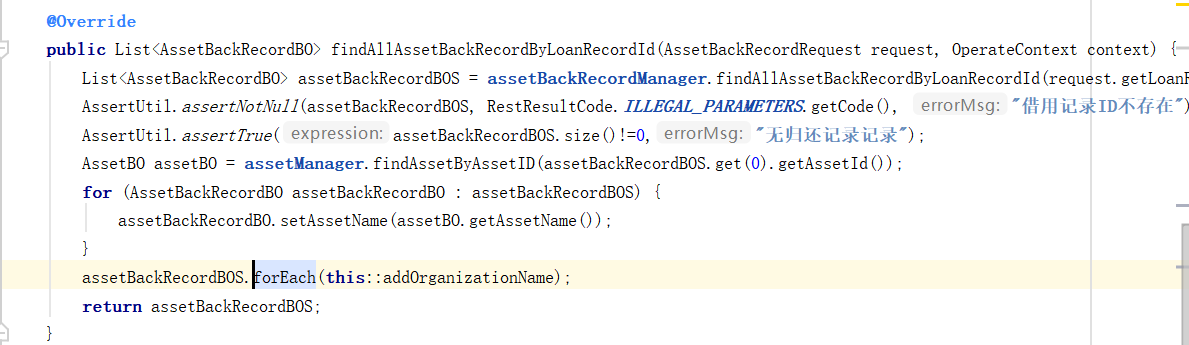
Lambda 表达式，也可称为闭包，它是推动 Java 8 发布的最重要新特性。

Lambda 允许把函数作为一个方法的参数（函数作为参数传递进方法中）。

使用 Lambda 表达式可以使代码变的更加简洁紧凑。

以下是lambda表达式的重要特征:

* **可选类型声明：**不需要声明参数类型，编译器可以统一识别参数值。
* **可选的参数圆括号：**一个参数无需定义圆括号，但多个参数需要定义圆括号。
* **可选的大括号：**如果主体包含了一个语句，就不需要使用大括号。
* **可选的返回关键字：**如果主体只有一个表达式返回值则编译器会自动返回值，大括号需要指定明表达式返回了一个数值。



**3.2.4通用工具层**

1. 总体介绍:

在后端开发中,如进行异常处理的功能在多个模块中均有涉及.重复复制粘贴代码代码,造成系统臃肿且不易修改.因此很多开发者会使用第三方开源工具类,如IOUtils等等,但是这些工具类提供基础功能,无法解决项目中独特需求,这也会导致在项目中已使用工具类,但要进行拓展,使工具类方法整体分离,不利于他人对代码的理解.另外所有的工具类都会为了适用各种情况,有很多对部分开发者来说完全无用的冗余代码.除此之外,有些特殊的功能,使用工具类反而会比较复杂.因此在后端开发中,我们使用自定义编写工具类,对功能类风格和代码规范都做了一个自己的定义,这样就能使得工具类完美贴合此项目开发,根据后端特定功能进行功能提取封装.减少了系统中重复代码冗余,降低之后子项目开发时间。

1. 异常处理工具

异常处理:是编程语言或计算机硬件里的一种机制，用于处理软件或信息系统中出现的异常状况,即超出程序正常执行流程的某些特殊条件。

使用自定义异常类统一对外异常展示方式，对于参数校验,非法数据输入可根据需要提前抛出自定义异常结束.同时自定义异常可根据项目实际需求弥补java中未定义异常.另一方面,使用自定义异常继承相关的的异常来抛出处理后的异常信息可以隐藏底层的异常,对于系统来说更加安全,异常信息也更加的直观.

1. 断言

断言:程序调试(测试代码或者调试程序时，总会做出一些假设，断言就是用于在代码中捕捉这些假设 ),断言就是程序中的一条语句，它对一个boolean表达式进行当程序已经处于不正确的状态，系统给出警告或退出程序.断言机制使Java程序避免if-else或switch语句进行变量状态检查,使检查的数据类型不同,语句形式仍然统一。

1. 数据格式返回统一工具

项目采用前后端分离:便于对接，提高项目开发速度,降低沟通成本,故定义内部统一接口数据返回格式(包括返回的状态码，提示信息，和需要的数据)则将其封装为统一格式是十分有必要的.

第四部分 第二课堂开发工具

4.1 Mysql

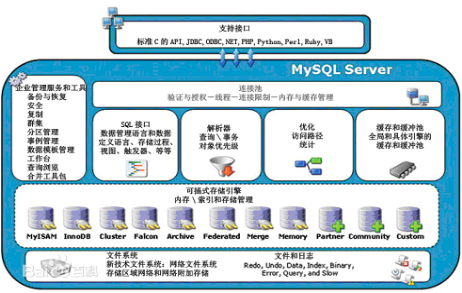


图 1‑31 MySQL架构及应用

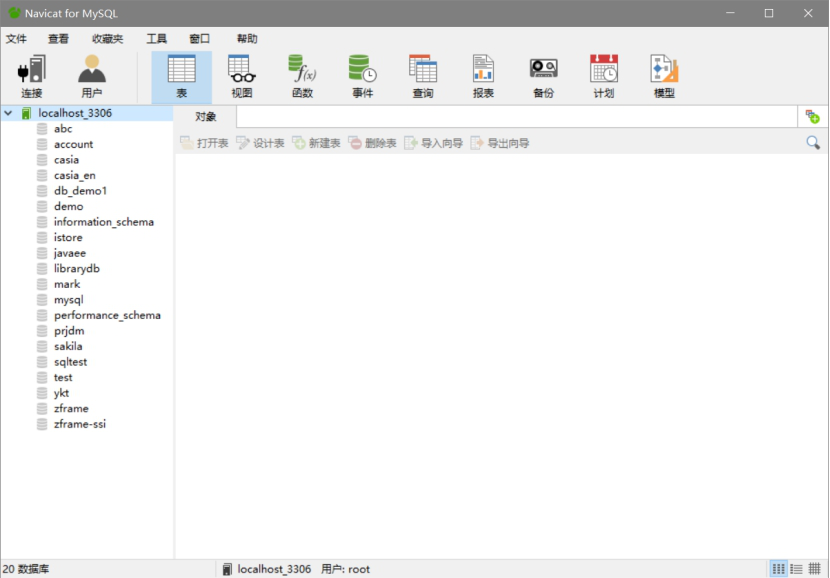
在本项目的开发过程中，我们主要利用MySQL和Navicat for MySQL两大软件进行数据库管理的相关操作。

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件。

MySQL是一种关系数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

4.2 Navicat



**图 3‑20 MySql可视化界面**

Navicat for MySQL是一款强大的 MySQL 数据库管理和开发工具，它为专业开发者提供了一套强大的足够尖端的工具，但对于新用户仍然易于学习。Navicat for MySQL 基于Windows平台，为 MySQL 量身订作，提供类似于 MySQL 的用管理界面工具。可执行增删改查，上传下载表数据等一系列可视化操作。

4.3 Idea



我们使用IDEA软件实现后台开发，IDEA 全称 IntelliJ IDEA，是java编程语言开发的集成环境。IntelliJ在业界被公认为最好的java开发工具之一，尤其在智能代码助手、代码自动提示、重构、J2EE支持、各类版本工具(git、svn等)、JUnit、CVS整合、代码分析、 创新的GUI设计等方面的功能可以说是超常的。而在本项目中我们主要运用IDEA软件实现Spring Boot框架的构建。

而在此之前，我们需要安装JDK。JDK是 Java 语言的软件开发工具包，主要用于移动设备、嵌入式设备上的java应用程序。JDK是整个java开发的核心，它包含了JAVA的运行环境，JAVA工具和JAVA基础的类库。

此外在主流web服务器Tomcat\**JBoss|WebSphere 我们选择tomcat,**Tomcat是Apache软件基金会（Apache Software Foundation）的Jakarta项目中的一个核心项目，由Apache、Sun和其他一些公司及个人共同开发而成。由于有了Sun的参与和支持，最新的Servlet 和JSP规范总是能在Tomcat中得到体现，Tomcat 8支持最新的Servlet 3.1和JSP 2.3规范。因为Tomcat技术先进、性能稳定，而且免费，因而深受Java爱好者的喜爱并得到了部分软件开发商的认可，是目前比较流行的Web应用服务器。

**同时**spring boot默认web程序启用tomcat内嵌容器tomcat,便于软件开发和环境配置

4.4 Webstrom

**第五部分 第二课堂功能展示**

**5.1第二课堂总体描述**

**5.2第二课堂功能描述**

**5.2.1校园活动**

a获取学生活动记录

b扫码取得活动章

c活动创建

d活动管理

e活动分配扫码员

**5.2.2财务系统**

a查看组织列表

b查看社团财务

c用户提交预算

d社团财务审核预算

e社团财务提交核算

**5.2.3物资管理系统**

a物资创建

b物资借用

c归还/报损

d物资管理

**5.2.4后台管理**

a后台主页

b物资创建

c财务处理

d物资管理

e讲座活动