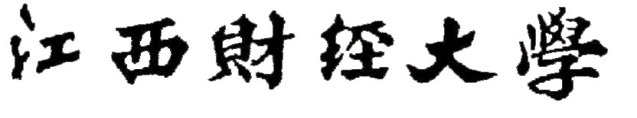
****

**本科学生设计性实验报告**

项 目 组 长 ： 张亮

成 员 ： 无

专 业 班 级 ： 软件192班

实验项目名称： Class Int-基于互联网的

教室资源管理系统数据库设计

指 导 教 师 ： 边海容

开 课 学 期 ： 2020-2021 学年 第 二 学期

**一、实验设计方案**

|  |  |  |
| --- | --- | --- |
| 实验名称：Class Int-基于互联网的教室资源管理系统数据库设计 | |  |
| 小组合作：是○否√ | 小组成员： 无 | |
| 1. 实验目的   通过数据库系统的设计，使学生了解数据库系统的详细设计过程，掌握数据库系统的设计方法，包括需求分析方法、模式设计方法、数据库的建立与维护，运用PL/SQL程序设计语言实现数据库系统的相关功能，培养学生在系统设计中多个不同角色中实际动手能力与相互配合、相互协作的能力。 | | |
| 1. 实验场地及仪器、设备和材料   实验场地：图文信息楼H104机房  实验设备：MacBook pro 笔记本一台，操作系统 macOS Monterey 12.0 beta  Windows 腾讯云服务器一台，操作系统 Windows server 2012  TDSQL-C 腾讯云企业分布式数据库一台，兼容 MySQL 5.7  实验软件：IntellijJ IDEA Ultimate Release、Navicat Premium、Draw.io  实验语言：Html、JavaScript、Java、SQL  实验框架：jQuery Mobile+Spring Boot2.5+Sping Data JPA | | |
| 1. 实验思路（实验内容、数据处理方法及实验步骤等）   以小组合作的形式，通过对给定项目进行需求分析与设计，完成数据库设计过程，运用某一种语言进行嵌入式SQL程序设计，最终实现一个能够演示的系统实现，并提交设计性实验报告。步骤：   1. 系统需求分析与数据字典、数据流图的编写 2. 完成数据库概念模式的抽象，设计系统的视图 3. 完成数据库系统的结构设计与物理设计 4. 完成需求分析中涉及到的部分功能 5. 对系统进行测试 | | |
| 指导老师对实验设计方案的意见  指导老师签名： | | |

**二、实验结果**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1.需求介绍**1.1项目前景 近年来，随着高校不断的扩招，在校师生人数不断增加，各级高校都迫切需要提高工作质量和工作效率。计算机信息处理技术发展的同时，也带动了网络技术的飞速发展，所有这些技术都为包括信息采集、信息处理、信息传递、信息共享功能的高度自动化的办公系统提供了强大的技术支持和开发平台，使得他们的开发和应用速率得到了很大的提供，各个系统的开发都成为可能。  教室是高等学校的教学管理中的重要资源，授课必须选择相关的教室才可进行。每学期开学前，各大高校都会组织老师安排指令性的教学任务，但与此同时，教室也要负责组织学校的各种其他活动，如讲座、培训等工作。教室活动的安排的增多，就需要对教室资源进行合理地安排，使教室的资源得到充分利用，这就是教室管理工作的主要任务。进入二十一世纪，互联网技术迅猛发展，很多高校都将教室管理进行网络化，就是在网络上进行相关的教室管理操作。通过网络，可以实现教室管理的快捷和方便。教室管理网络化的出现，构建了一个网上平台，供教室使用者和教室管理者交流。在教室使用者方便的查询和使用教室信息的同时，教室管理者也可以通过计算机对教室信息实现最有效率的管理，也就降低了人工处理的成本，管理人员也会更加轻松地从事相关的管理工作。  高校教室信息量大，动态变化频繁，要求可以精确、及时的对变化进行相应调整，所以教室管理是一个复杂的过程，管理员需要设定专门的数据库，并及时地更新和完善相关的教室信息库，要达到这个目的，最简单的方法就是建立教室管理系统。通过这个系统，可以大大提高高校信息化建设，可以在提高教室管理工作的效率和质量的同时，为学校制定相关的决策提供必要的依据。  本文从小组开发的并投入使用的荟庐报告厅微信借用小程序入手，尝试从小到大，从学院到学校，从几个教室到全校教室，进行“Class Int——基于互联网的教室资源管理系统”的项目搭建。  教室的使用具有一定的计划性和流动性，即一个班在相应的时间里，所上的课程和使用的教室是固定的，但是不同的课程在不同的时间里会使用不同的教室，这与教务处的课程安排是密切相关的。  为了能够更系统的，更有序的，更合理的，更有效地进行教室管理，有必要利用计算机来处理各种信息，这也就需要一个更有效的教室管理系统。  运用软件工程的基本原理和方法应用，对多媒体教室管理系统进行需求分析、系统架构、模块划分等提出具体的解决方案。  本系统是对教室的使用情况进行管理，为用户提供了一套操作简单、使用可靠、界面友好、易于管理和使用的处理工具。本系统对教室使用情况进行统一处理，避免数据存取、数据处理的重复，提高工作效率，减少了系统数据处理的复杂性。本系统不仅使管理人员从繁重的工作中解脱出来，而且提高了教室管理的效率，提高了教室管理的科学性，方便了用户查询、管理人员进行管理。  本系统是基于多媒体教室管理工作的需求、结合学生需求开发的多媒体教室管理系统，因此该系统结构清晰，简单实用，可以满足教学的需求。操作人员一般不用培训就能使用该系统。通过这样的教室管理系统，可以做到教室的集中化、规范化管理，实现准确、快速查询统计功能，从而减少教学人员的工作量，大大缩短了师生预约教室的时间。将全面提升教室管理的信息化管理水平，提高学校管理人员、教学人员的工作效率，降低学校的管理成本，提升教室的使用率，为学生进行科研设计、校内活动提供便利，为学校创造更大的社会效益。 1.2需求分析 本系统的最终用户为在校师生，我们根据从学校方面取得的图表资料、文字资料以及其他细节方面的信息，根据我们日常生活中的经验，根据我们所做的其他询问和调查，得出用户的下列实际要求： **1.2.1学校的组织机构情况** 与教室管理相关的学校的组织机构有：学生、教师和教室。学校的所有日常工作都是主要围绕着这三大部分进行的。  学生方面，一个学校下设若干学院，如软件学院、VR现代产业学院、信管学院等；一个学院下设若干专业，如软件学院下设三个专业：软件工程、物联网工程等；一个专业有若干班级，如软件学院的2019级软件工程专业下设六个班级：软件191班、软件192班、软件193班、软件194班、软件195班、软件196班、；一个班级有若干同学，如软件192班有张亮、李婷等。  教师方面，一个学校下设若干学院，如软件学院、VR现代产业学院、信管学院等；一个学院下设若干专业，如软件学院下设三个专业：软件工程、物联网工程、软件工程（中外合作）；一个专业有若干教师，如软件工程专业有边海容老师、陈积富老师等。  课程方面，一个学校下设若干学院，如软件学院、VR现代产业学院、信管学院等；一个学院下设若干专业，如软件学院下设三个专业：软件工程、物联网工程、软件工程（中外合作）；一个专业开设有若干课程，如软件工程专业开设有数据库系统原理、面向对象建模与分析等。  教室方面，一个学校有若干教学楼，如一教、二教、群庐、英庐、荟庐、萃庐等；一个教学楼有若干楼层，如荟庐教学楼有一层、二层、三层、四层、五层；一个楼层有若干教室，如一层有W101,W102,W103,W104等教室。  一个教师可以开设若干门课程，一门课程可以由多个老师来教授。一个教室在不同的时段可以上不同的课程。 **1.2.2调查相关部门的业务活动情况** 教务处：  教务处需要处理借教室申请信息，使用的数据是电子版的教室、教师和学生信息，对提出的借用教室申请采用手动的加工和处理，最后给借教室的人输出的是批准或拒绝借用教室的信息，即一个电子的教室使用条。 **1.2.3用户对系统的要求** 信息要求：  由于系统的使用主体是教师和学生，因此对系统的信息要求可分为以下几个方面：  a、教师信息  教师的基本信息，主要包括教师的教师编号，教师姓名，所属院系，职称，身份证号等；  b、学生信息  学生的基本信息，主要包括学生的学生编号，学生姓名，所属院系，职务，身份证号等；  c、教室信息  教室的基本信息，主要包括教室的教室编号，教学楼号，楼层号，多媒体设备配备情况等；  d、教学楼信息  教学楼的基本信息，主要包括教学楼名称，教学楼编号等；  e、课程信息  课程的基本信息，主要包括课程名称、课程时间段等；  f、预约信息表  预约的相关信息，包括预约人、预约时间、预约描述、联系方式等；  处理要求：  学校现存系统存在的问题：  a、教务系统：  我们从自身体验出发，并通过问卷调查了许多老师同学，认为学校现存的借用教室的程序过于繁琐，浪费时间，并且给教务处老师带来了很大的工作量，仅能在工作时间进行审核。  现在我校教务平台可以查阅每个教室当天各个时段的使用情况：“有课”、“无课”。但是，每一页显示的时间过短、借用时间不灵活、借用限制多、审核时间长、不能查询任意教室在任意时段的使用情况、不能通过手机操作，完全不利于用户查询和借用。  b、物业楼管：  由于教务系统的各种限制，使得近半数老师学生更倾向于向楼管人员口头进行登记借用的方式，相比于教务系统，有审核快、时间灵活的优势，同时也带来冲突、需要当面借用等一系列缺点。  鉴于以上存在的各种问题，给学校的主体――教师和学生，带来了很大的不便，使得现存的教务系统不能更好地服务于教师和同学，也不利于教务处审核的老师。因此我们认为有必要设计新系统，完善上述各种功能。  系统应当完成以下的信息处理：  a、教室查询；  学生或者老师通过这个功能，可以通过手机APP、小程序等客户端查询相关教学楼相关教室的信息以及该教室在每天任一时段的使用情况，例如有课、举办讲座、举行活动等等。这个功能以便使大家能更好地了解教室及其使用情况。  b、教室借用；  学生或者老师通过这个功能，可以通过手机APP、小程序等客户端借教室，即获得教室在某段时间的使用权，办讲座，开展社团活动等等。  c、借用审核；  教务处老师通过这个功能，可以通过手机APP、小程序等客户端实时接收借用申请并进行审核。  安全性要求：  a、系统应设置访问用户的标识以鉴别是否是合法用户，并要求合法用户设置其密码，保证用户身份不被盗用；  b、系统应对不同的数据设置不同的访问级别，限制访问用户可查询和处理数据的类别和内容；  c、系统应对不同用户设置不同的权限，区分不同的用户，该系统的用户主要可以分为以下几类：  ●普通用户：该类用户主要由学生、教职工等组成。用户可以使用微信授权使用本系统客户端小程序，正常使用教室的查看、借用、分享、调整、续约、举报等功能。同时对于部分高级用户（部分被授权的老师、学生），提供数据导出、举报查看等功能。  ●审核管理人员：该类用户主要组成为学校教务处信息管理科相关管理人员。在拥有普通用户、高级用户所有权限的同时，可以对所有借用申请进行审核，也可以发布公告。  ●系统管理员：主要操作信息管理系统，承担数据登入登出，修改，备份等工作，确保信息不泄漏、不丢失，采取一切可能的技术手段和管理措施，保护网络中的信息安全。  ●运维人员：软件的测试维护人员。针对在系统的日常运营中可能出现的服务器过载，宕机，软件本身出现的bug等问题进行预防和解决。  ●合作人员：平台的合作伙伴，该软件可能需要经学校网络管理中心同意，调取微信用户对应的个人信息，如学号、姓名等，用于破坏教室追责、实名认证等。  主要功能模块主要为以下几类：  ●公共模块：任何用户均可以访问以下界面；  （1）主界面：系统主要功能界面；  （2）注册界面：用户需要注册账号后登录；  （3）登录界面：用户注册成功后，需要登录或者授权登陆系统才可以拥有访问其他页面的权限。  ●用户模块：不同的用户对应访问不同的界面的权限；  普通用户（学生、普通老师）：  （1）教室详情模块：用户在该页面查看教室使用情况，以进一步进行借用预约、续约、等操作；  （2）预约详情模块：用户在该页面可以对自己的预约信息进行查看、修改等操作；  （3）聊天模块：用户在该页面可以对某个预约的发起人、系统客服发起私聊操作；  （4）帮助模块：用户在该页面可以查看系统的使用说明，对遇到的问题可以选择私聊系统客服、提交反馈、提交等操作；  （5）个人界面：用户在该页面可以查看自己的所有预约，同时对个人信息、权限等进行设置。  高级用户（被授权学生、被授权老师、相关教职工、物业楼管人员）：  除包含以上普通用户的所有模块以外，高级用户还拥有以下功能：  （1）数据导出界面：用户在该界面导出教室使用的基本情况；  （2）教室详情模块：在原有的基础上，该类用户可以查看申请者的实名信息；  （3）审核举报模块：用户在该界面可以查看举报信息，并进行相关扣分、意见反馈等操作。  ●管理员模块：管理相关信息，不同的管理员类型对应访问不同的界面的权限；  审核管理员用户（教务处相关老师）：  除包含高级用户的特有模块以外，高级用户还拥有以下功能：  （1）预约审核模块：用户在该界面查看并审核相关预约，并进行同意预约或者拒绝预约等操作；  （2）授权权限模块：审核管理员可以赋予指定用户高级用户权限，使其能够访问高级用户功能。  系统管理员用户（系统开发维护人员）：  （1）用户信息管理模块：可以对所有用户信息进行查看、修改、删除以及授予高级用户、管理员权限操作；  （2）备份恢复模块：可以对数据库信息进行备份、恢复操作。   1. 完整性要求：   a、各种信息记录的完整性，信息记录内容尽量不为空；  b、各种数据间相互的联系的正确性；  c、相同的数据在不同记录中的一致性。 **1.2.4确定系统的边界** 经对前面的需求调查和初步的分析，确定由计算机完成的工作时对数据进行各种管理和处理，具体的工作内容见第二部分。由手工完成的工作主要有不能由计算机生成的，各种数据的更新，包括数据变化后的修改，数据的增加，失效数据或无用数据的删除等；以及系统的日常维护。 **2.技术描述** 2.1系统相关技术介绍**2.1.1系统开发相关技术**  1. 系统总体架构   前后端分离已成为互联网项目开发的业界标准使用方式，通过nginx+tomcat的方式，也可以中间加一个nodejs有效的进行解耦，并且前后端分离会为以后的大型分布式架构、弹性计算架构、微服务架构、多端化服务（多种客户端，例如：浏览器，车载终端，安卓，IOS等等）打下坚实的基础。这个步骤是系统架构从猿进化成人的必经之路。核心思想是前端HTML页面通过AJAX调用后端的API接口并使用JSON数据进行交互。  Web服务器：一般指像Nginx这类的服务器，他们一般只能解析静态资源；  应用服务器：一般指像Tomcat，Jetty，Resin这类的服务器可以解析动态资源也可以解析静态资源，但解析静态资源的能力没有web服务器好；  一般都是只有web服务器才能被外网访问，应用服务器只能内网访问。  以前的Java Web项目大多数都是Java程序员既搞前端，又搞后端。随着时代的发展，渐渐的许多大中小公司开始把前后端的界限分的越来越明确，前端工程师只管前端的事情，后端工程师只管后端的事情。正所谓术业有专攻，一个人如果什么都会，那么他毕竟什么都不精。大中型公司需要专业人才，小公司需要全才，但是对于个人职业发展来说，前后端需要分离。  早期主要使用MVC框架，Jsp+Servlet的结构图如下：  IMG_256  图1 Jsp+Servlet结构图  所有的请求都被发送给作为控制器的Servlet，它接受请求，并根据请求信息将它们分发给适当的JSP来响应。同时，Servlet还根据JSP的需求生成JavaBeans的实例并输出给JSP环境。JSP可以通过直接调用方法或使用UseBean的自定义标签得到JavaBeans中的数据。需要说明的是，这个View还可以采用 Velocity、Freemaker 等模板引擎。使用了这些模板引擎，可以使得开发过程中的人员分工更加明确，还能提高开发效率。  这种方式耦合性太强。那么，就算你用了freemarker等模板引擎，不能写Java代码。那前端也不可避免的要去重新学习该模板引擎的模板语法，无谓增加了前端的学习成本。  因此，我们需要前后端半分离模式，前端负责开发页面，通过接口（Ajax）获取数据，采用Dom操作对页面进行数据绑定，最终是由前端把页面渲染出来。这也就是Ajax与SPA应用（单页应用）结合的方式，其结构图如下：  IMG_256  图2 前后端分离结构图  步骤如下：  （1）浏览器请求，CDN返回HTML页面；  （2）HTML中的JS代码以Ajax方式请求后台的Restful接口；  （3）接口返回Json数据，页面解析Json数据，通过Dom操作渲染页面；  后端提供的都是以JSON为数据格式的API接口供Native端使用，同样提供给WEB的也是JSON格式的API接口。  那么意味着WEB工作流程是：  1、打开web，加载基本资源，如CSS，JS等；  2、发起一个Ajax请求再到服务端请求数据，同时展示loading；  3、得到json格式的数据后再根据逻辑选择模板渲染出DOM字符串；  4、将DOM字符串插入页面中web view渲染出DOM结构；  首先，这种方式的优点是很明显的。前端不会嵌入任何后台代码，前端专注于HTML、CSS、JS的开发，不依赖于后端。自己还能够模拟Json数据来渲染页面。发现Bug，也能迅速定位出是谁的问题。 **2.1.2端开发相关技术**  1. Html5   HTML是互联网上应用最广泛的标记语言。HTML文件就是普通文本+HTML标记，而不同的HTML标记能表示不同的效果。（简单的说HTML是超文本标记语言）  HTML5草案的前身名为 Web Applications 1.0，于2004年被WHATWG提出，于2007年被W3C接纳，并成立了新的 HTML工作团队。  如果从狭义的角度来讲，HTML5就是HTML4的新一代产品。  而如果从广义的角度来讲，则是新一代的富客户端解决方案。  HTML5的优势在于支持Html5的浏览器包括Firefox（火狐浏览器），IE9及其更高版本，Chrome（谷歌浏览器），Safari，Opera等；国内的 遨游浏览器（Maxthon），以及基于IE或Chromium（Chrome的工程版或称实验版）所推出的360浏览器、搜狗浏览器、QQ浏览器、猎豹浏览器等国产浏览器同样具备支持HTML5的能力。  需要注意的是，虽然很多浏览器目前已经能够支持HTML5，但是显示效果仍旧存在差异性。   1. JavaScript   JavaScript一种动态类型、弱类型、基于原型的客户端脚本语言，用来给HTML网页增加动态功能。  动态：在运行时确定数据类型。变量使用之前不需要类型声明，通常变量的类型是被赋值的那个值的类型。  弱类：计算时可以不同类型之间对使用者透明地隐式转换，即使类型不正确，也能通过隐式转换来得到正确的类型。  原型：新对象继承对象（作为模版），将自身的属性共享给新对象，模版对象称为原型。这样新对象实例化后不但可以享有自己创建时和运行时定义的属性，而且可以享有原型对象的属性。  JavaScript由三部分组成：  1.ECMAScript（核心）  作为核心，它规定了语言的组成部分：语法、类型、语句、关键字、保留字、操作符、对象  2.DOM（文档对象模型）  DOM把整个页面映射为一个多层节点结果，开发人员可借助DOM提供的API，轻松地删除、添加、替换或修改任何节点。DOM也有级别，分为DOM1、DOM2、DOM3，拓展不少规范和新接口。   1. 3. BOM （浏览器对象模型）   支持可以访问和操作浏览器窗口的浏览器对象模型，开发人员可以控制浏览器显示的页面以外的部分。   1. 4.JavaScript版本   JavaScript语言是在10天时间内设计出来的，虽然语言的设计者水平非常NB，但谁也架不住“时间紧，任务重”，所以，JavaScript有很多设计缺陷。  此外，由于JavaScript的标准——ECMAScript在不断发展，最新版ECMAScript 6标准（简称ES6）已经在2015年6月正式发布了，所以，讲到JavaScript的版本，实际上就是说它实现了ECMAScript标准的哪个版本。   1. JavaScript   jQuery Mobile是一个易于触摸的Web UI开发框架，可让您开发可在智能手机和平板电脑上使用的移动Web应用程序。 jQuery Mobile框架建立在jQuery核心之上，并提供了许多功能，包括HTML和XML文档对象模型（DOM）遍历和操纵，处理事件，使用Ajax执行服务器通信以及网页的动画和图像效果。 。 移动框架本身是从jQuery核心单独下载的大约12KB（压缩和压缩），压缩/压缩后约为25KB。 与其他jQuery框架一样，jQuery Mobile是一个免费的双重许可（MIT和GPL）库。  尽管jQuery Mobile仍在Alpha中，但仍有一些演示和文档。 建议您查看的文档和演示相关信息 ，并期待在演示源代码下载部分 。  在撰写本文时，jQuery Mobile框架是Alpha 2版本（v1.0a2）。 本守则为草案形式，可能会随时更改。 但是，现有框架非常可靠。 有了alpha版本中令人印象深刻的组件集，jQuery Mobile有望成为开发移动Web应用程序的理想框架和工具集。  该框架易于使用。 您可以主要使用很少或没有JavaScript的标记驱动来开发页面。  尽管jQuery Mobile利用最新HTML5，CSS3和JavaScript，但并非所有移动设备都提供这种支持。 jQuery Mobile的理念是同时支持高端和功能较弱的设备（例如不支持JavaScript的设备），并仍提供最佳体验。  jQuery Mobile在设计时考虑了可访问性。 它支持可访问的富Internet应用程序（WAI-ARIA），以帮助使用辅助技术的残障游客访问网页。  jQuery Mobile框架的整体大小相对较小，JavaScript库为12KB，CSS为6KB，还有一些图标。  该框架还提供了一个主题系统，使您可以提供自己的应用程序样式。  当工具箱，例如PhoneGap的使用（参见相关信息 ），它使用网络技术来构建独立的应用程序，jQuery Mobile框架可以帮助简化您的应用程序的开发。 **2.1.3后端开发相关技术**  1. Spring   Spring框架是J2EE应用开发的集成解决方案，提供了IOC（控制反转）和AOP（面向切面）两种核心机制，为应用程序内部各模块之间实现高内聚、低耦合提供了支持。IOC，又称“控制反转”，是一种根据配置实例化Java对象，管理对象生命周期，组织对象之间关系的设计思想。Spring框架将纳入生命周期管理的Java对象称之为”Bean”，Spring框架在启动时自动创建Bean，并将Bean放到Spring的上下文中。如果某个Bean申明需要关联另外一个Bean，Spring框架自动建立Bean之间的关联。当某个Bean申明需要关联另外一个Bean时，可以申明关联另外一个Bean的接口，Spring会自动从上下文中查找实现该接口的Bean，从而建立两者之间的关联。在IOC机制的支持下，Spring可以J2EE体系中各种技术集成起来，如图所示。    图3 JSpring结构图  这些技术包含Web开发技术（SpringWebMVC）、数据持久化技术（SpringORM）、缓存技术（SpringDataCache）、Restful客户端（SpringRestTemplate）、安全技术（SpringSecurity）、服务注册发现和负载均衡（SpringCloud）。Spring支持各种组件存在不同的第三方实现方案，这些第三方实现方案并可相互替换，开发者可根据场景选择最适合的实现方案，当需要修改实现方案时，仅需要对应用进行简单的配置，不需要对已完成的代码做任何改动。比如，数据缓存技术（SpringDataCache）存在将数据缓存到redis、缓存到memcache、缓存到本地内存几种方案，开发者只需要调用缓存API，而不需要关注具体实现。再比如，服务注册发现和负载均衡框架（SpringCloud）框架体系中，需要搭建服务注册中心，服务注册中心的实现技术有etcd、consul、eureka、dubbo等，这些实现技术来自不同的公司或开源组织，而开发者选择或切换技术实现时，仅需要简单的配置，无需修改代码。  AOP，又称面向切面编程。面向切面思想从面向对象思想基础上发展而来，用于将系统的核心功能和辅助功能解耦。web设计开发者在设计系统的某一功能模块时，除了要设计该功能本身的逻辑实现，还需要考虑其辅助功能，如记录日志、进行权限控制、对数据进行缓存、对调用方进行流量控制等等。Spring将上述辅助功能看作“切面”，切面是一个独立的模块，调用者调用服务提供者的API的过程会透明触发切面的代码逻辑，切面负责对调用请求进行拦截、处理、过滤。   1. Spring Boot   SpringBoot为基于J2EE架构的web后端集成开发框架。SpringBoot从Spring框架发展而来，在Spring框架的基础上，简化Spring框架的默认配置，如支持在应用程序中嵌入web服务器实现可独立运行的web应用，从而简化web应用的部署。   1. Spring Data Java Persistent API   Java Persistent API（Java 数据持久化API）简称JPA[28]。Java 是一种面向对象的编程语言，信息在Java 应用内存中是以类和对象的形式组织的，对象拥有属性、方法和关联关系。而企业的生产运营数据通常由数据库管理，数据库按存储方式，可以分为关系型数据库、keyvalue数据、列式数据库、图形数据库等。关系型数据库是企业生产应用的主流数据库，其按照表、字段、约束的形式组织数据结构，应用程序通过SQL（结构化查询语言）操作关系型数据库的数据。  良好的系统架构设计应具备数据独立性特征，即数据结构的改变不影响上层的应用程序，数据独立性包含物理独立性和逻辑独立性两个方面。物理独立性表示数据磁盘等介质的存储结构的改变不影响应用程序，表现为底层数据库中间件的变动对应用程序透明，如将Oracle更换为MYSQL 或其他数据库。逻辑独立性表示数据逻辑结构的变化对应用程序透明，如增加表、增加字段。JPA 定义了Java 应用程序和关系型数据库之间的接口，具体功能有：  定义了对Java对象新增、修改、删除、查询接口，应用程序逻辑仅需要面向JPA 编程。  通过元数据定义Java 对象、属性、关系和关系型数据库表、字段、约束之间的映射，将面向对象的API 翻译成可由数据库执行的SQL 语句。  JPA 实现了数据的物理独立性。如JPA 提供了对不同关系数据库dialect（方言）的支持，实现同一个API 针对不同的关系数据库产品，翻译成不同的SQL。如分页查询A 表，每页10行，查询第1 页的场景，针对MYSQL 生成的SQL 是“select \* from A limit 0,10”，而针对Oracle 的语法却是“select \* from (select rownum rownum\_ a.\* from A a where rownum<=10) whererownum\_>=1”。  JPA 实现了数据的逻辑独立性。关系数据库数据模型变动后，需要调整Java 对象和表、字段、约束的映射的元数据映射，对上层应用代码透明。  JPA 按照接口和实现相分离的原则设计，具备较强的可扩展性，JPA 定义了一套API 标准，由第三方团队实现此标准。应用程序的开发者可选择JPA 的实现，更改JPA 实现对上层应用代码无任何影响。   1. 腾讯云分布式数据库TDSQL-C   云原生数据库 TDSQL-C（Cloud Native Database TDSQL-C，TDSQL-C）是腾讯云自研的新一代高性能高可用的企业级分布式云数据库。融合了传统数据库、云计算与新硬件技术的优势，100%兼容 MySQL 和 PostgreSQL，实现超百万级 QPS 的高吞吐，128TB 海量分布式智能存储，保障数据安全可靠。  TDSQL-C基于MySQL5.7，相比于mysql，具有以下特点：  定制内核  深度定制的数据库内核，实现诸多企业级特性和优化，服务公司内部用户和腾讯云百 TB 级别的外部用户，是支撑关键业务平稳运行的基石。  日志即数据库  可计算智能存储，由分布式存储系统自动管理数据的多副本，实现自动扩缩容，自动故障校验检测和修复。日志即数据库，真正实现了将 Redo LOG 下沉到存储层，将网络 IO 减少到最低。  面向服务的体系结构  架构基于现有的云服务如对象存储 COS，云硬盘 CBS，云服务器 CVM，云网络服务如私有网络 VPC，腾讯网关服务 TGW（Tencent Gateway）。  软件优化与新硬件相结合  通过基于 SPDK 和 RDMA 的零拷贝技术，减少了操作系统上下文切换以及数据在用户态和内核态之间拷贝引起的性能损耗，进一步优化了关键路径的系统性能，降低请求延迟。 2.2系统功能的设计和划分 根据如上得到的用户需求，我们将本系统按照所完成的功能分成以下四个子系统： **2.2.1用户信息子系统** a、用户点击登陆、注册后，用户需要填写相关信息，并提交登陆/注册/在验证成功后进入，并赋予相关权限。  b、用户和系统其他部分可以通过该系统对用户的部分个人信息进行读取访问和修改。 **2.2.2事件消息子系统** 用户进入问题反馈、举报、设备报修、教室借用详情页面、私聊消息后，用户需要填写完整后提交数据库。在提交相关事件后，经过系统处理发送给相关人员进行进一步审核/核查并提交结果返回给用户。 **2.2.3教室处理子系统** 用户进行教室借用时，该系统提取数据库中教室的相关数据，经过提取并格式化后传输给客户端并呈现在用户面前以便于进一步借用操作。  管理员更新教室时，可以通过上传学校的开课表，该系统通过分析、提取、格式化开课表内的相关数据后将更新的相关信息写入数据库。 **2.2.4日志消息子系统** 用户和其他系统以及程序本身运行时所做的操作、产生的信息和错误都会被该系统所捕获，经过汇总处理和格式化后输出至相关数据库，便于管理运维人员进行系统维护。  经上述分析，我们已经得到了对于该系统的基本要求和系统模块的划分，综上，我们对教师查询子系统、教室借用子系统进行具体的数据库设计，在需求分析中形成的数据流图如下一部分所示。 2.3数据流图 数据流图（Data Flow Diagram）简称DFD，它从数据传递和加工角度，以图形方式来表达系统的逻辑功能、数据在系统内部的逻辑流向和逻辑变换过程，是结构化系统分析方法的主要表达工具及用于表示软件模型的一种图示方法。  为了表达处理过程的数据加工情况，需要采用层次结构的数据流图。按照系统的层次结构进行逐步分解，并以分层的数据流图反映这种结构关系，能清楚和理解整个系统。 **2.3.1顶层数据流图（见图4）** 0层  图4 顶层数据流图 **2.3.2中间层数据流图（见图5）** /Users/zhangliang/Downloads/1层.png1层  图5 中间层数据流图 **2.3.3底层教室处理系统数据流图（见图6）** /Users/zhangliang/Downloads/3.1.png3.1  图6 底层教室处理系统数据流图 **2.3.4底层消息处理系统数据流图（见图7）** /Users/zhangliang/Downloads/3.2.png3.2  图7 底层消息处理系统数据流图 **2.3.5底层中央监视系统数据流图（见图8）** /Users/zhangliang/Downloads/3.3.png3.3  图8 底层中央监视系统数据流图 2.4数据词典 数据字典是系统中各类数据描述的集合，是进行详细的数据收集和数据分析所获得的主要成果。通常包括：数据项、数据结构、数据流、数据存储和处理过程五个部分。  数据字典是对数据流图的详细描述。  结合上一部分所做数据流图，对该系统的数据词典部分进行设计与分析。   1. 用户信息数据词典   表1 数据词典描述·用户ID   |  |  | | --- | --- | | 数据项编号 | I 01 01 | | 数据项名称 | 用户ID | | 数据项字段 | id | | 数据项描述 | 用于系统辨别用户 | | 数据项类型 | Int(15) | | 数据项备注 | 主键、自增 |   表2 数据词典描述·用户姓名   |  |  | | --- | --- | | 数据项编号 | I 01 02 | | 数据项名称 | 用户姓名 | | 数据项字段 | name | | 数据项描述 | 用于储存用户姓名 | | 数据项类型 | varchar(10) | | 数据项备注 | 非空 |   表3 数据词典描述·用户用户名   |  |  | | --- | --- | | 数据项编号 | I 01 03 | | 数据项名称 | 用户用户名 | | 数据项字段 | user\_name | | 数据项描述 | 用于储存用户用户名 | | 数据项类型 | varchar(20) | | 数据项备注 | 非空 |   表4 数据词典描述·用户密码加密盐   |  |  | | --- | --- | | 数据项编号 | I 01 04 | | 数据项名称 | 用户密码加密盐 | | 数据项字段 | salt | | 数据项描述 | 用于用户密码加密 | | 数据项类型 | varchar(16) | | 数据项备注 | 非空，注册时随机生成16位字符串，对原始用户密码进行非对称矩阵加盐加密 |   表5 数据词典描述·用户密码   |  |  | | --- | --- | | 数据项编号 | I 01 05 | | 数据项名称 | 用户密码 | | 数据项字段 | password | | 数据项描述 | 用于储存用户密码 | | 数据项类型 | varchar(255) | | 数据项备注 | 非空，用户经过非对称矩阵加盐加密算法后的密码 |   表6 数据词典描述·用户绑定手机号   |  |  | | --- | --- | | 数据项编号 | I 01 06 | | 数据项名称 | 用户绑定手机号 | | 数据项字段 | phone | | 数据项描述 | 用于储存用户手机号 | | 数据项类型 | int(11) | | 数据项备注 | 便于信息登记以及密码找回 |   表7 数据词典描述·用户绑定邮箱   |  |  | | --- | --- | | 数据项编号 | I 01 07 | | 数据项名称 | 用户绑定邮箱 | | 数据项字段 | email | | 数据项描述 | 用于储存用户邮箱 | | 数据项类型 | varchar(40) | | 数据项备注 | 便于密码找回 |   表8 数据词典描述·用户QQ openID   |  |  | | --- | --- | | 数据项编号 | I 01 08 | | 数据项名称 | 用户QQ openID | | 数据项字段 | qq | | 数据项描述 | 用于系统第三方快速登录 | | 数据项类型 | varchar(255) | | 数据项备注 |  |   表9 数据词典描述·用户微信 openID   |  |  | | --- | --- | | 数据项编号 | I 01 09 | | 数据项名称 | 用户微信 openID | | 数据项字段 | wechat | | 数据项描述 | 用于系统第三方快速登录 | | 数据项类型 | varchar(255) | | 数据项备注 |  |   表10 数据词典描述·用户苹果 openID   |  |  | | --- | --- | | 数据项编号 | I 01 10 | | 数据项名称 | 用户苹果 openID | | 数据项字段 | apple | | 数据项描述 | 用于系统第三方快速登录 | | 数据项类型 | varchar(6255) | | 数据项备注 |  |   表11 数据词典描述·用户一卡通号   |  |  | | --- | --- | | 数据项编号 | I 01 11 | | 数据项名称 | 用户一卡通号 | | 数据项字段 | stu\_id | | 数据项描述 | 用于信息登记 | | 数据项类型 | int(10) | | 数据项备注 | 以财大为例，一卡通号为10位 |   表12 数据词典描述·用户类型   |  |  | | --- | --- | | 数据项编号 | I 01 12 | | 数据项名称 | 用户类型 | | 数据项字段 | type | | 数据项描述 | 用于记录用户身份（学生/教师/审核员/管理员） | | 数据项类型 | varchar(5) | | 数据项备注 | 非空 |   表13 数据词典描述·用户高级权限   |  |  | | --- | --- | | 数据项编号 | I 01 13 | | 数据项名称 | 用户高级权限 | | 数据项字段 | high\_permissions | | 数据项描述 | 用于记录用户是否具有高级权限 | | 数据项类型 | bit | | 数据项备注 | 非空，0为false，1为true，默认值为0 |   表14 数据词典描述·用户认证   |  |  | | --- | --- | | 数据项编号 | I 01 14 | | 数据项名称 | 用户认证 | | 数据项字段 | authentication | | 数据项描述 | 用于记录用户是否认证 | | 数据项类型 | bit | | 数据项备注 | 非空，0为false，1为true，默认值为0 |   表15 数据词典描述·用户注册时间   |  |  | | --- | --- | | 数据项编号 | I 01 15 | | 数据项名称 | 用户注册时间 | | 数据项字段 | created\_date | | 数据项描述 | 用于记录用户注册时间 | | 数据项类型 | datetime | | 数据项备注 | 非空，无具体意义，用于生成用户加密盐 |   表16 数据词典描述·用户（密码）更新时间   |  |  | | --- | --- | | 数据项编号 | I 01 16 | | 数据项名称 | 用户（密码）更新时间 | | 数据项字段 | updated\_date | | 数据项描述 | 用于记录用户上次修改密码时间 | | 数据项类型 | datetime | | 数据项备注 | 非空，用于判断用户自动登录是否有效 |   表17 数据词典描述·用户上次登陆时间   |  |  | | --- | --- | | 数据项编号 | I 01 17 | | 数据项名称 | 用户上次登陆时间 | | 数据项字段 | login\_date | | 数据项描述 | 用于记录用户上次登陆时间 | | 数据项类型 | datetime | | 数据项备注 | 非空，用于判断用户当前登录是否有效 |   表18 数据词典描述·用户通知id   |  |  | | --- | --- | | 数据项编号 | I 01 18 | | 数据项名称 | 用户通知id | | 数据项字段 | server\_key | | 数据项描述 | 用于记录用户微信通知服务key | | 数据项类型 | datetime | | 数据项备注 | 用于向用户微信推送消息 |   表19 数据词典描述·用户设备ua   |  |  | | --- | --- | | 数据项编号 | I 01 19 | | 数据项名称 | 用户设备ua | | 数据项字段 | ua | | 数据项描述 | 用于记录用户设备 | | 数据项类型 | datetime | | 数据项备注 | 验证自动登陆是否有效 |   表20 数据词典描述·用户信用   |  |  | | --- | --- | | 数据项编号 | I 01 20 | | 数据项名称 | 用户信用 | | 数据项字段 | credit | | 数据项描述 | 用于记录用户信用值 | | 数据项类型 | Int(3) | | 数据项备注 | 非空，用于记录用户信用值 |   表21 数据词典描述·用户当前登陆令牌   |  |  | | --- | --- | | 数据项编号 | I 01 21 | | 数据项名称 | 用户当前登陆令牌 | | 数据项字段 | token | | 数据项描述 | 用于记录用户令牌 | | 数据项类型 | Int(3) | | 数据项备注 | 用于检测是否登陆有效 |   b、教室信息数据词典  表22 数据词典描述·教室ID   |  |  | | --- | --- | | 数据项编号 | I 02 01 | | 数据项名称 | 教室ID | | 数据项字段 | room\_id | | 数据项描述 | 用于系统辨别、储存教室名称 | | 数据项类型 | varchar(5) | | 数据项备注 | 主键，例如H104 |   表23 数据词典描述·教室所属教学楼   |  |  | | --- | --- | | 数据项编号 | I 02 02 | | 数据项名称 | 教室所属教学楼 | | 数据项字段 | room\_building | | 数据项描述 | 用于储存教室所属教学楼 | | 数据项类型 | varchar(10) | | 数据项备注 | 非空 |   表24 数据词典描述·教室所属楼层   |  |  | | --- | --- | | 数据项编号 | I 02 03 | | 数据项名称 | 教室所属楼层 | | 数据项字段 | room\_floor | | 数据项描述 | 用于储存教室所属楼层 | | 数据项类型 | varchar(4) | | 数据项备注 | 非空 |   表25 数据词典描述·教室最大容纳人数   |  |  | | --- | --- | | 数据项编号 | I 02 04 | | 数据项名称 | 教室最大容纳人数 | | 数据项字段 | room\_number | | 数据项描述 | 用于储存教室最大容纳人数 | | 数据项类型 | Int(3) | | 数据项备注 | 非空 |   表26 数据词典描述·教室类型   |  |  | | --- | --- | | 数据项编号 | I 02 05 | | 数据项名称 | 教室类型 | | 数据项字段 | room\_type | | 数据项描述 | 用于辨别教室类型 | | 数据项类型 | varchar(10) | | 数据项备注 | 非空，例如：机房、多媒体教室、普通教室 |   c、事件消息信息数据词典  表27 数据词典描述·事件ID   |  |  | | --- | --- | | 数据项编号 | I 03 01 | | 数据项名称 | 事件ID | | 数据项字段 | event\_id | | 数据项描述 | 用于系统辨别事件 | | 数据项类型 | int | | 数据项备注 | 主键、自增 |   表28 数据词典描述·事件类型   |  |  | | --- | --- | | 数据项编号 | I 03 02 | | 数据项名称 | 事件类型 | | 数据项字段 | event\_type | | 数据项描述 | 用于储存事件类型 | | 数据项类型 | varchar(10) | | 数据项备注 | 非空，例如：预约、反馈、举报等 |   表29 数据词典描述·事件标题   |  |  | | --- | --- | | 数据项编号 | I 03 03 | | 数据项名称 | 事件标题 | | 数据项字段 | event\_title | | 数据项描述 | 用于简述事件 | | 数据项类型 | varchar(20) | | 数据项备注 | 非空 |   表30 数据词典描述·事件描述   |  |  | | --- | --- | | 数据项编号 | I 03 04 | | 数据项名称 | 事件描述 | | 数据项字段 | event\_des | | 数据项描述 | 用于描述事件 | | 数据项类型 | varchar(255) | | 数据项备注 |  |   表31 数据词典描述·事件开始时间   |  |  | | --- | --- | | 数据项编号 | I 03 05 | | 数据项名称 | 事件开始时间 | | 数据项字段 | event\_s | | 数据项描述 | 用于储存事件开始时间 | | 数据项类型 | datetime | | 数据项备注 | 非空 |   表32 数据词典描述·事件结束时间   |  |  | | --- | --- | | 数据项编号 | I 03 06 | | 数据项名称 | 事件结束时间 | | 数据项字段 | event\_e | | 数据项描述 | 用于储存事件结束时间 | | 数据项类型 | datetime | | 数据项备注 | 非空 |   表33 数据词典描述·发起用户ID   |  |  | | --- | --- | | 数据项编号 | I 03 07 | | 数据项名称 | 发起用户ID | | 数据项字段 | user\_id | | 数据项描述 | 记录事件发起人 | | 数据项类型 | int | | 数据项备注 | 外键，对应用户信息表中的id |   表34 数据词典描述·涉及教室ID   |  |  | | --- | --- | | 数据项编号 | I 03 08 | | 数据项名称 | 涉及教室ID | | 数据项字段 | wechat | | 数据项描述 | 用于记录事件涉及教室 | | 数据项类型 | int | | 数据项备注 | 外键，对应教室信息表中的id |   表35 数据词典描述·事件已读   |  |  | | --- | --- | | 数据项编号 | I 03 09 | | 数据项名称 | 事件已读 | | 数据项字段 | read | | 数据项描述 | 用于记录事件是否已读 | | 数据项类型 | int | | 数据项备注 |  |   d、日志信息数据词典  表36 数据词典描述·日志ID   |  |  | | --- | --- | | 数据项编号 | I 04 01 | | 数据项名称 | 日志ID | | 数据项字段 | id | | 数据项描述 | 用于辨别系统日志 | | 数据项类型 | int | | 数据项备注 | 主键、自增 |   表37 数据词典描述·日志详情   |  |  | | --- | --- | | 数据项编号 | I 04 02 | | 数据项名称 | 日志详情 | | 数据项字段 | formatted\_message | | 数据项描述 | 控制台输出内容 | | 数据项类型 | text | | 数据项备注 | 非空 |   表38 数据词典描述·日志标题   |  |  | | --- | --- | | 数据项编号 | I 04 03 | | 数据项名称 | 日志标题 | | 数据项字段 | logger\_name | | 数据项描述 | 控制台输出标题 | | 数据项类型 | varchar(255) | | 数据项备注 | 非空 |   表39 数据词典描述·日志等级   |  |  | | --- | --- | | 数据项编号 | I 04 04 | | 数据项名称 | 日志等级 | | 数据项字段 | level\_string | | 数据项描述 | 用于储存该日志的等级 | | 数据项类型 | varchar(255) | | 数据项备注 | 非空，例如info、warn、error |   表40 数据词典描述·出错线程名   |  |  | | --- | --- | | 数据项编号 | I 04 05 | | 数据项名称 | 出错线程名 | | 数据项字段 | thread\_name | | 数据项描述 | 用于储存出错线程名 | | 数据项类型 | varchar(255) | | 数据项备注 | 尽可能获取，方便问题排查 |   表41 数据词典描述·调用文件   |  |  | | --- | --- | | 数据项编号 | I 04 06 | | 数据项名称 | 调用文件 | | 数据项字段 | caller\_filename | | 数据项描述 | 用于储存调用文件名称 | | 数据项类型 | varchar(255) | | 数据项备注 | 日志输出所在的java文件，方便问题排查 |   表42 数据词典描述·调用函数   |  |  | | --- | --- | | 数据项编号 | I 04 07 | | 数据项名称 | 调用函数 | | 数据项字段 | caller\_class | | 数据项描述 | 用于储存调用函数名称 | | 数据项类型 | varchar(255) | | 数据项备注 | 日志输出所在函数，方便问题排查 |   表43 数据词典描述·调用方法   |  |  | | --- | --- | | 数据项编号 | I 04 08 | | 数据项名称 | 调用方法 | | 数据项字段 | caller\_method | | 数据项描述 | 用于储存调用方法 | | 数据项类型 | varchar(255) | | 数据项备注 | 日志输出所在方法，方便问题排查 |   表44 数据词典描述·调用行   |  |  | | --- | --- | | 数据项编号 | I 04 09 | | 数据项名称 | 调用行 | | 数据项字段 | caller\_line | | 数据项描述 | 用于储存调用行 | | 数据项类型 | int | | 数据项备注 | 日志输出所在行数，方便问题排查 |  2.5数据结构定义 数据结构是计算机存储、组织数据的方式。数据结构是指相互之间存在一种或多种特定关系的数据元素的集合。本章将对系统主要的数据结构进行分析和定义。  表45 数据结构定义·用户账户信息   |  |  | | --- | --- | | 数据结构编号 | DS01 | | 数据结构名称 | 用户账户信息 | | 数据结构简述 | 用户登陆、注册信息描述 | | 数据结构组成 | I 01 01+I 01 02+I 01 03+I 01 04+I 01 07+I 01 08+I 01 01+I 01 12+I 01 13+I 01 14 |   表46 数据结构定义·用户个人信息   |  |  | | --- | --- | | 数据结构编号 | DS02 | | 数据结构名称 | 用户个人信息 | | 数据结构简述 | 用户个人资料描述 | | 数据结构组成 | I 01 05+I 01 06+I 01 09+I 01 10+I 01 11+I 01 15 |   表47 数据结构定义·教室详情信息   |  |  | | --- | --- | | 数据结构编号 | DS03 | | 数据结构名称 | 教室详情信息 | | 数据结构简述 | 教室名称、信息描述 | | 数据结构组成 | I 02 01+I 02 02+I 02 03+I 02 04+I 02 05 |   表48 数据结构定义·事件详情信息   |  |  | | --- | --- | | 数据结构编号 | DS04 | | 数据结构名称 | 事件详情信息 | | 数据结构简述 | 事件类型、名称等描述 | | 数据结构组成 | I 03 01+I 03 02+I 03 03+I 03 04+I 03 05+I 03 06+I 03 07+I 03 08 |   表49 数据结构定义·日志信息   |  |  | | --- | --- | | 数据结构编号 | DS05 | | 数据结构名称 | 日志信息 | | 数据结构简述 | 日志类型、名称等描述 | | 数据结构组成 | I 04 01+I 04 02+I 04 03+I 04 04+I 04 05+I 04 06+I 04 07+I 04 08+I 04 09 |  2.6数据流定义 数据流是一组有序，有起点和终点的字节的数据序列。包括输入流和输出流。  数据流最初是通信领域使用的概念，代表传输中所使用的信息的数字编码信号序列。  本部分将对系统主要的几个数据流进行定义。  表50 数据流定义·提交用户事件消息   |  |  | | --- | --- | | 数据流编号 | D1 | | 数据流名称 | 提交用户事件消息 | | 数据流简述 | 用户提交预约申请、反馈、举报等事件消息 | | 数据流来源 | 用户 | | 数据流去向 | 事件消息系统 | | 数据流组成 | I 03 01+I 03 02+I 03 03+I 03 04+I 03 05+I 03 06+I 03 07+I 03 08 |   表51 数据流定义·教室空闲数据   |  |  | | --- | --- | | 数据流编号 | D2 | | 数据流名称 | 教室空闲数据 | | 数据流简述 | 系统将处理后的空闲教室数据 | | 数据流来源 | 教室处理系统 | | 数据流去向 | 用户 | | 数据流组成 | I 02 01+I 02 02+I 02 03+I 02 04+I 02 05 |  2.7数据库设计与实现 在 B/S 结构的Web 系统中，数据库是整个系统的数据中心，用户所需的信息都是通过中间控件调用数据库数据而得到的。数据库的设计在本系统中占有很大的比重，一个良好的数据库不但可以使系统以较优秀的性能运行，也可以简化开发难度，缩短开发周期。本系统采用的数据库是MySQL 数据库，前文中提到了数据库设计的大体思路并给出了数据库的表结构和数据库表的关系表，下面就细节方面进一步细化。  概念数据模型的目标是统一业务概念，作为业务人员和技术人员之间沟通的桥梁，确定不同实体之间的最高层次的关系。概念数据模型是最终用户对数据存储的看法，反映了最终用户综合性的信息需求，它以数据类的方式描述企业级的数据需求，数据类代表了在业务环境中自然聚集成的几个主要类别数据。  概念模型设计阶段，主要处于系统分析的阶段，属性可以不完全描述，但也可以描述一些主要的属性。概念数据模型的内容包括重要的实体及实体之间的关系。在概念数据模型中不包括实体的属性，也不用定义实体的主键。这是概念数据模型和逻辑数据模型的主要区别。  /Users/zhangliang/Downloads/概念模型ER图2.0.png概念模型ER图2.0  图6 数据库概念模型E-R图  逻辑数据模型的目标是尽可能详细的描述数据，但并不考虑数据在物理上如何来实现，也是逻辑模型与物理模型之间区别性的关键。例如在本系统数据库的逻辑模型中可能加入了由于系统设计需要的一些字段（属性），这些字段可能是在业务概念上不存在或不需要的。逻辑模型是概念模型从真实世界向计算机世界的转换，加入了系统设计的相关内容。逻辑数据建模不仅会影响数据库设计的方向，还间接影响最终数据库的性能和管理。如果在实现逻辑数据模型时投入得足够多，那么在物理数据模型设计时就可以有许多可供选择的方法。逻辑数据模型反映的是系统分析设计人员对数据存储的观点，是对概念数据模型进一步的分解和细化。逻辑数据模型是根据业务规则确定的，关于业务对象、业务对象的数据项及业务对象之间关系的基本蓝图。逻辑数据模型的内容包括所有的实体和关系，确定每个实体的属性，定义每个实体的主键，指定实体的外键，需要进行范式化处理。接着上面逻辑结构的设计，该系统的的逻辑结构，如下图所示。  逻辑模型  图7 数据库逻辑模型图  物理数据模型的目标是指定如何用具体的数据库模式来实现逻辑数据模型，以及真正的保存数据。  物理数据模型是在逻辑数据模型的基础上，考虑各种具体的技术实现因素，进行数据库体系结构设计，真正实现数据在数据库中的存储。  物理数据模型的内容包括确定所有的表和列，定义外键用于确定表之间的关系，基于用户的需求可能进行范式化等内容。在物理实现上的考虑，可能会导致物理数据模型和逻辑数据模型有较大的不同。  物理模型跟逻辑模型的区别就是，逻辑模型并不指出特定的数据存储，仅限于系统逻辑上的描述。物理模型是逻辑模型在具体存储介质上的表现，直接与具体的数据库管理系统或存储介质相关的数据模型。  物理模型给出了在数据库系统的字段名称，与具体数据库管理系统相关的数据类型的定义。而逻辑模型与具体的数据库管理系统或存储介质无关，仅为使用计算机系统概念中的一种逻辑结构。  /Users/zhangliang/Downloads/event.pngevent  图8 数据库物理模型图  在数据库的设计当中，本系统尽量采用单表设计（即尽可能将近似数据存储在同一个数据表里）。相比于设计多个表，单表设计具有以下的好处：  a、从开发效率来看：  联合查询是需要多个单查询进行逻辑组合才能完成的查询的工作，联合查询仅仅需要一个SQL就可以完成查询工作，即把业务逻辑放到了SQL中，由数据库来处理，相对来说开发效率会比较高。  b、从查询效率来看：  查询的执行流程：连接数据库、传入SQL、执行SQL语句、返回查询结果、断开连接；  无论是单查询还是联合查询，进行查询时都是需要进行上述流程的。传统的实现中，认为需要让数据库来完成更多的工作，这样做的原因在于网络通信、查询解析和优化是一件代价很高的事情。然而现在的众多数据库在设计上连接和断开连接都是轻量级的，返回一组小的查询结果也很高效。并且现在的网络速度与之前相比也快了很多，连接数据库、返回查询结果、断开连接的耗时不在是影响效率的主要原因。那么SQL的执行耗时成了关键，多个单查询的耗时根据情况不同无法与联合查询的耗时进行对比，不过我们可以通过以下几个方面进行考虑：  1.缓存效率：  数据库是存在缓存机制的，当一条SQL执行之后，再次执行相同的SQL，数据库会把缓存的结果返回出去，而不会重新查询数据库。单查询的可重用性较高，所以缓存效率相较之联合查询会更高。使用第三方redis等缓存，key（组合更少更单一）和value使用也相应减少。  2.锁竞争：  为了保证数据库的数据同步，在数据库进行读写时，数据库会用锁机制，限制其他连接对其操作。读写越快，数据库的并发性越高。由于联合查询查询速度比单个查询要慢很多，这样联合查询会增加锁的竞争关系，所以用单查询会更好些。  3.查询结果有效使用率：  相较于联合查询，单查询的查询结果有效利用率要高很多，也就是说联合查询会浪费一些时间在查询无用的数据上。例如后台管理的列表界面，通常都会分页显示，关联查询的结果集，只有当前页的数据被使用，其他都是无用的，但数据库需要消耗额外资源得到全部结果集，再从中得到当前页数据。单表查询结果放redis等缓存中使用效率更高。  4.大数量的表推荐使用单表，小数据量的表推荐使用组合查询。  5.单表SQL虽然设计难度大但是简单容易理解，而且做分库等改动较小。  综合以上三个模型的设计与分析，本系统的表设计归纳总结如下图：  2021-06-27_16_13_05  2021-06-27_16_13_05  2021-06-27_16_13_05  2021-06-27_16_13_05  2021-06-27_16_13_05  图9 数据库系统表设计图 2.8主要系统功能模块设计与实现**2.8.1构建Web入口** 首先构建简单首页index.html，作为入口。利用渐入渐出的加载效果实现仿APP式的加载。构建了8张启动图并随机启用，同时在前端和后端效验用户登陆状态。如果效验成功，则直接进入app，否则继续判断。如果用户第一次使用该app，则会跳转到权限验证页面，告知用户相关权限，待用户同意后，再跳转至登陆页面。  相关js数据处理代码如下：   |  | | --- | | *checkLogin*()  tools.ajaxGet("http://localhost:8080/getInfo", autoUser, *function* (res) {  console.log(res)  *if* (res.status === 0) {  console.log("云端获取成功")  *registerJsonn*(res.data)  ajax = *true*;  } *else* {  console.log("云端认证失败")  document.getElementById('frame').src = "login.html";  document.title = "笑约";  ajax = *true*;  } }) |  **2.8.2构建登录前端页面** 接下来是登陆页面的搭建。采用js、jsp为辅css、html为主，构建页面。  动态背景采用纯css实现，backdrop-filter属性为一个元素后面区域添加模糊效果，使用filter(滤镜) 属性，改变颜色，hue-rotate(deg) 给图像应用色相旋转，calc() 函数用于动态计算长度值，var() 函数调用自定义的CSS属性值x，调用动画animate，需要10s完成动画，linear表示动画从头到尾的速度是相同的，infinite指定动画应该循环播放无限次，动态计算动画延迟几秒播放，部分代码如下：   |  | | --- | | .box .circle {  position: absolute;  background: rgba(255, 255, 255, 0.1);  backdrop-filter: blur(5px);  box-shadow: 0 25px 45px rgba(0, 0, 0, 0.1);  border: 1px solid rgba(255, 255, 255, 0.5);  border-right: 1px solid rgba(255, 255, 255, 0.2);  border-bottom: 1px solid rgba(255, 255, 255, 0.2);  border-radius: 50%;  filter: hue-rotate(calc(var(--x) \* 70deg));  animation: animate 10s linear infinite;  animation-delay: calc(var(--x) \* -1s);  }  @keyframes animate {  0%, 100% {  transform: translateY(-50px);  }  50% {  transform: translateY(50px);  }  } |   登录窗口代码，同时集成错误提示，实现一页多用，由前端JavaScript和后端API一同控制，部分代码展示：   |  | | --- | | <form *id*="box2" *style*="display:none">  <h1 *id*="logintext" *class*="boxtext">登陆成功</h1>  <div *class*="inputBox">  <input *type*="button" *value*="退出登录" *id*="tc">  </div>  <h1 *id*="returntext" *class*="boxtext" *style*="font-size: 24px"></h1>  </form> |   其中，box2为隐藏窗口。只有当登陆成功后，服务器API返回参数success并且cookie中的username和加密的token经过JavaScript校对后相对应，才能显示这个隐藏的box2（即登陆成功），同时将登陆窗口box1隐藏。box3为错误显示窗口，原理和作用同box2。不一样的是，JavaScript通过获取服务器返回的错误代码（详见后文），修改box3中的错误提示，达到提醒“账号密码错误”、“验证码错误”、“用户未登录”等效果。  验证码通过腾讯云API实时生成，并将验证码结果存在后端中供登陆验证核对验证码。同时对验证码增加监听事件，局部刷新验证码，达到点击切换验证码的效果。  form表单通过ajax实现访问登陆接口进行登陆验证。  使用type="checkbox"标签做为是否自动登陆的选择框，一起提交到后台Struts。  以下为部分js代码，实现一个网页多用，进行未登录提示、验证码错误提示、用户名密码错误提示、网页跳转，控制登陆窗口、成功窗口、报错窗口直接的显示以及隐藏：   |  | | --- | | } *else if* (msgcode.indexOf("zhuce") !== -1) {  box1.style.display = 'none'  box4.style.display = 'block'  logintxt.innerjsp = '欢迎注册' } *else if* (msgcode.indexOf("tc") !== -1) {*//退出登录* tc.onclick() } *else* { *//其他状态  //首先尝试进行自动登录  if* (*checkLogin*()) {  *jump*("success") *//自动登录成功* }  *//自动登录失败，继续解析参数  if* (msgcode === "uperror") { *//如果为账号密码错误状态* box1.style.display = 'none'  box3.style.display = 'block'  errortext.innerjsp = "用户名或者密码错误！"  logintxt.innerjsp = '登陆失败'  *let* i = 6  *setInterval*(*function* () { *//倒计时* i = i - 1  errorreturntext.innerjsp = i + "秒后返回登陆"  *if* (i === 0) {  *jump*()  }  }, 1000)  } *else if* (msgcode === "icodeerror") { *//如果为游客状态，提示用户先登录* errorreturntext.innerjsp = 6 + "秒后前往登陆"  box1.style.display = 'none'  box3.style.display = 'block'  errortext.innerjsp = "验证码错误！"  logintxt.innerjsp = '操作失败'  *let* i = 6  *setInterval*(*function* () {  i = i - 1  errorreturntext.innerjsp = i + "秒后返回登陆"  *if* (i === 0) {  *jump*()  }  }, 1000)  } *else if* (msgcode === "youke") { *//如果为游客状态，提示用户先登录* errorreturntext.innerjsp = 6 + "秒后前往登陆" |   当登陆按钮被点击后，触发form表单通过ajax提交get请求给api进行读取，包括name、pwd以及确验证码，实现获取相关信息，便于后续的操作。   |  | | --- | | *let* captcha1 = *new* TencentCaptcha('2004603663', *function* () {  tools.ajaxGet("http://localhost:8080/login", loginuser, *function* (res) {  console.log(res)  *var* jsonObject = JSON.parse(res.data)  *if* (res.status === 0) {  *if* (chck.value === "true") {  *setCookie*("loginName", *btoa*(*noTuiGe*(jsonObject.username)), "7",  "/") *//登陆用户名被编码保存，一方面防止中文存入cookie崩溃，一方面"加密"  setCookie*(*btoa*(*btoa*(*noTuiGe*(jsonObject.username))), *jiamitoken*(jsonObject.username), "7",  "/") *//登陆用户名被两次编码进行"加密"，同时保存对应的加密token* } *else* {  *setCookie*("loginName", *btoa*(*noTuiGe*(jsonObject.username)), "/")  *setCookie*(*btoa*(*btoa*(*noTuiGe*(jsonObject.username))), *jiamitoken*(jsonObject.username), "/")  }  *jump*("success")  } *else  jump*("uperror")  }) }); |   后端和数据库数据以及验证码进行验证。   |  | | --- | | @RequestMapping("login") *public* ResResult login(  @RequestParam String password,  @RequestParam String userName,  *HttpServletResponse* response,  *HttpServletRequest* request) {  *ajax*(response);  User user;  asyncLogs.sendInfoAsync("用户登陆" + userName + "/" + password);  *try* {  user = userRepository.findByUserName(userName);  } *catch* (IncorrectResultSizeDataAccessException e) {  System.***out***.println(userName + "用户重复");  *return* ResResult.*fail*(ResultCode.***USER\_HAS\_EXISTED***);  } *catch* (NullPointerException e) {  System.***out***.println(userName + "用户不存在");  *return* ResResult.*fail*(ResultCode.***USER\_NOT\_EXISTED***);  }  *if* (user == *null*) {  System.***out***.println(userName + "用户不存在");  *return* ResResult.*fail*(ResultCode.***USER\_NOT\_EXISTED***);  }  System.***out***.println(user);  *if* (!encryption.*getSaltMD5*(password, user.getSalt()).equals(user.getPassword()))  *return* ResResult.*fail*(ResultCode.***USER\_PWD\_ERROR***);  JSONObject object = *new* JSONObject(*true*);  object.put("status", "登陆成功");  object.put("username", user.getUserName());  object.put("name", user.getName());  user.setLoginDate(*new* Date());  user.setUa(request.getHeader("User-Agent"));  asyncUserRepo.saveAsync(user);  System.***out***.println("返回数据");  *return* ResResult.*suc*(object.toString()); } |   登陆分为以下三种情况：  验证码错误，直接返回页面，并携带参数 msg=icerror 和 登陆前页面的url（后文会提到），被前端js读取后，显示验证码错误提示信息，并6秒后返回登陆页面重试。  验证码正确，账户和密码也正确，登陆成功。同时，将登陆信息生成token；将用户名直接写入Cookie便于前端读取显示到网页；将用户名、登陆时间、token经base64加密后写入Cookie，防止被修改，用于验证自动登录。最后携带参数 msg=success 和登陆前页面的url返回登陆页面，被前端js读取后，显示XXX用户登陆成功提示信息，并6秒后返回登陆前页面。如果登陆前页面不存在，则返回首页（index.jsp）。  其中，若选中自动登录，则cookie将会保存一个星期，以供自动登录。一周后，自动登录也将会随着cookie的消除而失效。  验证码正确，但账号和密码至少其一有误，登陆失败，携带参数 msg=icerror 和 登陆前页面的url（后文会提到），被前端js读取后，显示账号密码错误提示信息，并6秒后返回登陆页面重试。   |  | | --- | | *if* (realCode != *null* && realCode.equals(userCode)) {  *if* ("zhangliang".equals(name) && "123".equals(pwd)) {  session.setAttribute("isLogin", *true*);  Cookie coo = *new* Cookie("LOGINTOKEN", MyBase64.*base64jiami*("user=" + name + ";time=" + *new* SimpleDateFormat("yyyy/MM/dd-HH:mm:ss:SSS").format(*new* Date()) + ";自动登录:" + autologin));  Cookie coon = *new* Cookie("ursename", name);  *if* ("true".equals(autologin)) {  coon.setMaxAge(60 \* 60 \* 24 \* 7);  coo.setMaxAge(60 \* 60 \* 24 \* 7);*//设置cookie的生命周期* System.*out*.println("自动登录");  }  response.addCookie(coo);*//添加到response中* response.addCookie(coon);  response.sendRedirect("loginPage.jsp?msg=success&from=" + fromUrl);} *else* {response.*sendRedirect*("loginPage.jsp?msg=uperror&from=" + fromUrl);System.*out*.println("账号密码错误");  } } *else* {  System.*out*.println("验证码错误");  *//request.getRequestDispatcher("/loginPage.jsp").forward(request, response);* response.sendRedirect("loginPage.jsp?msg=icerror&from=" + fromUrl); } |   其中，用户的敏感信息被加密为token后保存在浏览器中保证安全。  同时后端还实现了注册接口，采用异步储存来加快速度。相关底层实现如下：   |  | | --- | | @Async *public void* saveAsync(User user) {  save(user); }  *public void* save(User user) {  System.***out***.println("异步储存");  userRepository.save(user); } |   SpringData自带的Repository 接口：CrudRepository 接口提供了最基本的对实体类的添删改查操作 - T save(T entity);保存单个实体 - T findOne(ID id);根据id查找实体 - void delete(ID/T/Iterable);根据Id删除实体，删除实体，批量删除 PagingAndSortingRepository提供了分页与排序功能 - <T, ID extends Serializable>第一个参数传实体类，第二个参数传注解数据类型 - Iterable<T> findAll(Sortsort);排序 - Page<T> findAll(Pageable pageable);分页查询（含排序功能） JpaSpecificationExecutor提供了Specification(封装 JPA Criteria查询条件)的查询功能 - List<T> findAll(Specification<T> spec); - Page<T>findAll(Specification<T> spec, Pageable pageable); - List<T> findAll(Specification<T> spec, Sortsort);  这里值列出的是常用方法。  CrudRepository 中的findAll() 方法要慎用。当数据库中数据量大，多线程脚本调用findAll方法，系统可能会宕机。  CrudRepository 中的deletAll()方法要慎用。这是物理删除，现在企业一般采用逻辑删除。  PagingAndSortingRepository 和JpaSpecificationExecutor 能满足大部分业务需求。  同时，在修改密码的接口上，提供的接口已无法满足使用，上面的方法虽然简单(不用写sql语句)，但它有最为致命的问题-----不支持复杂查询，其次是命名太长因此可以采用一下办法：  使用@Query 注解实现复杂查询，设置 nativeQuery=true使查询支持原生sql  配合@Modifying 注解实现创建，修改，删除操作  SpringData 默认查询事件为只读事务，若要修改数据则需手动添加事务注解。  查询方法名一般以 find | read | get 开头，建议用find findByAccount : 通过account查询User account是User的属性，拼接时首字母需大写。支持的关键词有很多比如 Or,Between,isNull,Like,In等，findByEmailEndingWithAndCreatedDateLessThan : 查询在指定时间前注册，并以xx邮箱结尾的用户 And : 并且 EndingWith :以某某结尾 LessThan : 小于。注意 若有User(用户表) Platform(用户平台表) 存在一对一的关系，且User表中有platformId字段 SpringData 为了区分：findByPlatFormId 表示通过platformId字段查询 findByPlatForm\_Id 表示通过platform实体类中id字段查询  表的设计，尽量做单表查询，以确保高并发场景减轻数据库的压力。  例如这里提到的通过邮箱或者手机号模糊查询用户信息:   |  | | --- | | @Query(value = "SELECT u FROM User u WHERE u.email LIKE %?1% OR u.phone LIKE %?2%") *List*<User> findByEmailAndIhpneLike(String email, String iphone); |  **2.8.3构建软件首页** 进入项目首页时，将会再次对用户登陆身份进行验证，同时向api请求个人信息：   |  | | --- | | *let* user = *atob*(*getCookie*("loginName")) *//尝试获取用户名cookie let* autoUser = {  'userName': user } *let* result; tools.ajaxGet("http://localhost:8080/getInfo", autoUser, *function* (res) {  console.log(res)  *if* (res.status === 0) {  console.log("云端获取成功")  *registerJsonn*(res.data)  *let* name = *getUserName*();  userObj = *getUser*(name)  *if*(userObj.type.indexOf("审核")!==-1) {  document.getElementById("fun2").innerHTML="预约审核"  document.getElementById("fun2e").innerHTML="Appointment review"  document.getElementById("noEvent").innerHTML="审核拒绝"  document.getElementById("noEvent1").style.display="inline-block";  document.getElementById("noEvent").classList.remove("kefu")  document.getElementById("noEvent1").classList.remove("kefu")  document.getElementById("noEvent").classList.add("shenheB")  document.getElementById("noEvent1").classList.add("shenheB")}  *updateInfo*()  } }) |   ajax异步调用后端接口，获取到信息后写入到本地储存中以减少不必要的api请求，同时将信息动态显示在网页上，局部更新。  AJAX 即“Asynchronous Javascript And XML”（异步 JavaScript 和 XML），是指一种创建交互式网页应用的网页开发 技术。  ajax 是一种浏览器通过 js 异步发起请求，局部更新页面的技术。  Ajax 请求的局部更新，浏览器地址栏不会发生变化 局部更新不会舍弃原来页面的内容  局部更新：就是页面上的某个组件 如div中的值进行了更新数据。 **2.8.4前后端JSON数据交互及标准API返回格式** JSON(JavaScript Object Notation) 是一种轻量级的数据交换格局。它基于ECMAScript的一个子集。 JSON选用完全独立于言语的文本格局，但是也使用了类似于C言语宗族的习气（包含C、C++、C#、Java、JavaScript、Perl、Python等）。这些特性使json调试成为抱负的数据交换言语。 易于人阅览和编写，同时也易于机器解析和生成(一般用于提高网络传输速率)。  在学习JavaScript的过程中，我们接触了一种新的数据格式——JSON数据格式。JSON的全称是”JavaScript Object Notation”，意思是javascript对象表示法，它是一种基于文本，独立于语言的轻量级数据交换格式。 JSON 数据的书写格式是键（名称）/值对。 JSON 键值对是用来保存 JS 对象的一种方式，和 JS 对象的写法也大同小异，键/值对包括字段名称（在双引号中），后面写一个冒号，然后是值。 JSON 值可以是：字符串（在双引号中）、数组（在中括号中）、数字（整数或浮点数）、逻辑值（true 或 false）、对象（在大括号中）、 null。 JSON结构有两种结构，就是对象和数组。通过这两种结构可以表示各种复杂的结构。 {"province": "Shanxi"} 可以理解为是一个包含province为Shanxi的对象， ["Shanxi","Shandong"]这是一个包含两个元素的数组 而 [{"province": "Shanxi"},{"province": "Shandong"}] 就表示包含两个对象的数组。当然了,也可以使用 {"province":["Shanxi","Shandong"]} 来简化上面的JSON,这是一个拥有一个name数组的对象。  后端JSON的生成上，我们使用了fastjson，实现对象快速转换jsonobject  fastjson是阿里巴巴的开源JSON解析库，它可以解析JSON格式的字符串，支持将Java Bean序列化为JSON字符串，也可以从JSON字符串反序列化到JavaBean。  fastjson的优点有以下：  速度快  fastjson相对其他JSON库的特点是快，从2011年fastjson发布1.1.x版本之后，其性能已经被其他Java实现的JSON库超越。  使用广泛  fastjson在宏网宏大量使用，在万台服务器上部署，fastjson在业界被广泛接受。在2012年被中国读者大量使用为国产开源软件之一。  测试完备  fastjson 有非常多的测试用例，在 1.2.11 版本中，测试用例超过 3321 个。每次发布进行回归测试，保证质量稳定。  使用简单  fastjson的API十分简洁。  功能完备  支持泛型，支持流处理超大文本，支持枚举，支持序列化和反序列化扩展。  前端的使用上，js原生支持对json字符串和对象的相互转化。在ajax异步请求qpi获取到返回信息后，提起其中的json数据并进一步提取，从而进一步操作页面。   |  | | --- | | *for* (*let* i = keys.length-1; i >=0; i--) {  *let* keyname = (keys[i].toString())  valus.push(json[keyname])  *if* (json[keyname]["type"] === "审核中") {  div += '<li data-icon="false">\n' +  ...................... |   后端还进一步对返回数据进行API标准化，使之成为标准的JSON API返回格式。  为了兼容多种类型的错误码，我们通过声明接口的方式解决，再由具体的业务错误码类实现该接口。  首先在包中添加response目录并新建返回码接口类。其次再定义一个业务错误码枚举类实现上述接口类。继续在其它API中添加包并新建 Result 返回包装类。其中提供了 SuccessfulResult 及ErrorResult 方法用于接口调用成功或失败时的返回。  截屏2021-06-30 下午8.50.24 **2.8.5后端异步调用函数** Spring异步线程池的接口类，其实质是java.util.concurrent.Executor  Spring 已经实现的异常线程池：  SimpleAsyncTaskExecutor：不是真的线程池，这个类不重用线程，每次调用都会创建一个新的线程。  SyncTaskExecutor：这个类没有实现异步调用，只是一个同步操作。只适用于不需要多线程的地方  ConcurrentTaskExecutor：Executor的适配类，不推荐使用。如果ThreadPoolTaskExecutor不满足要求时，才用考虑使用这个类  SimpleThreadPoolTaskExecutor：是Quartz的SimpleThreadPool的类。线程池同时被quartz和非quartz使用，才需要使用此类  ThreadPoolTaskExecutor ：最常使用，推荐。 其实质是对java.util.concurrent.ThreadPoolExecutor的包装  spring对过@Async定义异步任务  异步的方法有3种。最简单的异步调用，返回值为void。带参数的异步调用，异步方法可以传入参数。异常调用返回Future。  对于不依赖返回数据的相关数据库操作，我们都都将其改为异步多线程函数，并发运行，极大提升相应速度：   |  | | --- | | *package* com.classint.xiaoyue.Async;  *import* com.classint.xiaoyue.data.event.Event; *import* com.classint.xiaoyue.data.event.*EventRepository*; *import* org.springframework.beans.factory.annotation.Autowired; *import* org.springframework.scheduling.annotation.Async; *import* org.springframework.stereotype.Component;  @Component *public class* AsyncEventRepo {  @Autowired *private EventRepository* eventRepository;   @Async  *public void* saveEventAsync(Event eve) {  save(eve);  }   *public void* save(Event eve) {  eventRepository.save(eve);  }   @Async  *public void* deleteAsync(*long* id) {  delete(id);  }   *public void* delete(*long* id) {  eventRepository.deleteEventById(id);  } } |  **2.8.6非对称加密算法** 为保护用户数据，在后端密码的保存上，我们采用了基于矩阵的加盐加密算法：  利用待加密信息配合成熟算法通过一定规则构造出一个新的加密矩阵,将待加密信息经过此加密矩阵转换为密文数据,将密文数据与原始数据传输到密文接收方,密文接收方利用接收到的明文信息采用相同规则构造出解密矩阵,将密文数据解密,再将解密信息与明文信息比较,如对比一致,则采用该信息。本发明所述的方法不但可以保证信息传输的安全性,而且能有效检测出信息是否被篡改。   |  | | --- | | *private static* String md5Hex(String str) {  *try* {  MessageDigest md = MessageDigest.*getInstance*("MD5");  *byte*[] digest = md.digest(str.getBytes());  *return new* String(*new* Hex().encode(digest));  } *catch* (Exception e) {  e.printStackTrace();  System.***out***.println(e);  *return* "";  } } |   所有密码数据均被加密后保存在数据库中，无法逆向进行破解。  截屏2021-06-30 下午9.08.24  即使在前端，我们也运用了不可逆非对称加盐加密算法来进行保密。一般信息也被层层编码储存在客户端浏览器中。  不可逆算法指加密过程中不需要使用密钥，输入明文后由系统直接经过加密算法处理成密文，这种加密后的数据是无法被解密的，只有重新输入明文，并再次经过同样不可逆的加密算法处理，得到相同的加密密文并被系统重新识别后，才能真正解密。这里使用不可逆加密算法MD5。MD5（Message-Digest Algorithm）是计算机安全领域广泛使用的散列函数（又称哈希算法、摘要算法）。主要用来确保消息的完整性和一致性。常见的应用场景有：密码保护、下载文件校验等。当攻击者知道算法是md5后，可以将实现准备好的常见明文密码的md5值来进行匹配暴力破解所以要进行"加盐"处理在密码的特定位置插入特定字符串后，再对修改的字符串进行md5运算同样的密码，当“盐”值不一样的时候，md5的值差异非常大通过密码加盐，可以防止最初级的暴力破解，如果攻击者事先不知道“盐”值，破解的难度就会非常大。   |  | | --- | | *let* salt = "HELLOhtmlImZHANGliang@5201314,<.>/?;:'[china]"  *function strToBinary*(str) {  *let* result = [];  *let* list = str.split("");  *for* (*let* i = 0; i < list.length; i++) {  *if* (i !== 0) {  result.push((salt[i % (salt.length - 1)]).charCodeAt(0).toString(2)); *//加入转换成二进制的盐* }  *let* item = list[i];  *let* binaryStr = item.charCodeAt(0).toString(3); *//原字符转为asc2后转换为三进制* result.push(binaryStr);  }  *return* result.join("110"); }  *function jiamipassword*(text) {  *return* md5(md5(*strToBinary*(text))) *//md5加密两次* }  *function jiamitoken*(text) {  *return* md5(text + "这是个验证用户的token") } |  **2.8.7项目云端部署及网络内容分发加速** 项目搭建在腾讯云轻量云服务器上，并采用腾讯云网络分发加速。  腾讯云轻量应用服务器（Lighthouse）是新一代开箱即用、面向轻量业务场景的云服务器产品，助力中小企业和开发者便捷高效的在云端构建小型网站、博客、论坛、云盘以及各类开发测试和学习环境，相比传统云服务器更加简单易用，并通过基础云资源与热门开源软件的融合打包实现应用的一站式交付。  内容分发网络（Content Delivery Network，CDN），是在现有 Internet 中增加的一层新的网络架构，由遍布全球的高性能加速节点构成。这些高性能的服务节点都会按照一定的缓存策略存储您的业务内容，当您的用户向您的某一业务内容发起请求时，请求会被调度至最接近用户的服务节点，直接由服务节点快速响应，有效降低用户访问延迟，提升可用性。  CDN 有效地解决了目前互联网业务中网络层面的以下问题：  用户与业务服务器地域间物理距离较远，需要进行多次网络转发，传输延时较高且不稳定。  用户使用运营商与业务服务器所在运营商不同，请求需要运营商之间进行互联转发。  业务服务器网络带宽、处理能力有限，当接收到海量用户请求时，会导致响应速度降低、可用性降低。  具体部署为：  首先需要将Spring Boot项目打包为War包：  war是一个可以直接运行的web模块，通常用于网站，打成包部署到容器中。以Tomcat来说，将war包放置在其\webapps\目录下，然后启动Tomcat，这个包就会自动解压，就相当于发布了。  war包是Sun提出的一种web应用程序格式，与jar类似，是很多文件的压缩包。war包中的文件按照一定目录结构来组织。根据其根目录下包含有html和jsp文件，或者包含有这两种文件的目录，另外还有WEB-INF目录。通常在WEB-INF目录下含有一个web.xml文件和一个classes目录，web.xml是这个应用的配置文件，而classes目录下则包含编译好的servlet类和jsp，或者servlet所依赖的其他类（如JavaBean）。通常这些所依赖的类也可以打包成jar包放在WEB-INF下的lib目录下。  简单来说，war包是JavaWeb程序打的包，war包里面包括写的代码编译成的class文件，依赖的包，配置文件，所有的网站页面，包括html，jsp等等。一个war包可以理解为是一个web项目，里面是项目的所有东西。  SpringBoot默认达成jar包，使用SpringBoot构想web应用，默认使用内置的Tomcat。但考虑到项目需要集群部署或者进行优化时，就需要打成war包部署到外部的Tomcat服务器中。修改pom.xml文件将默认的jar方式改为war，maven中排除spring-boot-starter-web中的Tomcat，添加打包依赖spring-boot-starter-tomcat。继承org.springframework.boot.web.servlet.support.SpringBootServletInitializer，实现configure方法。使用mvn命令行打包，运行：mvn clean和mvn install。  截屏2021-06-30 下午9.26.18  最后将war包部署至tomcat服务器。 **2.8.8数据库部署** 对于前文使用Data Grip实现的物理模型图，导入Navicat Data Modeler后直接转化成数据库。具体步骤为：  1.导入Navicat Data Modeler    2.选择同步至数据库    3.选择目标数据库。这里以“test”数据库为例。    4.确定操作后点击部署    5.检查无误后点击运行    6.操作完毕    6.查看结果：创建成功   **3.成果展示** 相关测试所需数据如下：  网页访问：  <https://xyzliang.jxufesoftware.club/ClassIntWeb/>  项目客户端：  笑约APP苹果系统版  笑约APP苹果手机版  （详见附件）  审核员账号（也可以在注册时输入11开头的江财一卡通号）  账户：zhangsan  密码：zl1424625705  学生账号（也可以在注册时输入非11开头的江财一卡通号）  账户1：xyzliang  密码1：zl1424625705  账户1：yeyilin  密码1：yeyilin  系统演示：  打开笑约APP或者网页，加载页面，有八个专门设计的加载图案。同时在进入和退出时渐入渐出，有一定的美观性。    授权页面，仅第一次进入网页显示。    注册页面，输入错误实时检测并显示，同时做了多设备适配保证显示效果。    注册后进入首页，简洁明了    进入预约页面——教学楼选择，简约大气的现代化UI设计    进入选择时间，当日已过或者无法预约时间将无法选中    进入详情页面，填写相关信息  截屏2021-06-30 下午9.51.19  更换管理员账号，进行预约审核      点击审核通过，切换回学生账号，即可看见审核成功  我的页面    个人资料页面    修改个人资料    返回查看，信息修改成功    根据系统状态，开启整体夜间模式 |

**三、实验结果分析**

|  |
| --- |
| ⑴本次实验成败之处及其原因分析  本小组虽然只有一人，但整体实验进度十分顺利，本次试验实现了本系统的基本功能，完成了需求分析阶段第一版的内容。  成功之处：  项目整体上，我们用SpringBoot和Spring Data进行后端API的开发以及数据库的访问，并采用腾讯云TQSQL—C分布式数据库储存系统数据，Web网页采用jQuery Mobile框架构建手机端网页。整个项目已经部署在腾讯云应用服务器上，配置了https协议以及CDN内容分发网络保证访问的快速以及安全。得益于b/S架构良好兼容性，我们可以随时把web app转化成实体的可安装在手机上的软件客户端。目前我们已经完成了对苹果手机、苹果电脑、windows电脑的客户端软件制作。  失败之处：  失败之处:时间安排不够合理，最后完成项目时间有些紧凑，还有部分功能没有实现，还需要对系统进一步修改完善。 |
| ⑵本实验的关键环节及改进措施  需求分析和数据库设计是本次实验的重中之重，数据流图和概念模型的设计过程应该全心思考分析,不断改进,数据结构应该 条理清晰,相互联系。将界面的功能进行优化扩展,用户使用界面进行美化,实现选课系统中数据 的可视化。 |
| 指导老师评语及得分：  签名：  年 月 日 |