

编 号: 2015214038

审定成绩: _____

重庆邮电大学

毕业设计（论文）

中文题目	基于二维码的考勤签到系统设计与实现
英文题目	Design and Implementation of Attendance Check-in System Based on Two-Dimensional Codes
学院名称	软件工程学院
学生姓名	黄卿怡
专 业	软件工程
班 级	13111505
学 号	2015214038
指导教师	代劲 教授
答 辩 组 负 责 人	贾朝龙 副教授

二〇一九年 六月

重庆邮电大学教务处制

_____学院本科毕业设计(论文)诚信承诺书

本人郑重承诺：

我向学院呈交的论文《_____》，是本人在指导教师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明并致谢。本人完全意识到本声明的法律结果由本人承担。

年级

专业

班级

承诺人签名

年 月 日

学位论文版权使用授权书

本人完全了解重庆邮电大学有权保留、使用学位论文纸质版和电子版的规定，即学校有权向国家有关部门或机构送交论文，允许论文被查阅和借阅等。本人授权重庆邮电大学可以公布本学位论文的全部或部分内容，可编入有关数据库或信息系统进行检索、分析或评价，可以采用影印、缩印、扫描或拷贝等复制手段保存、汇编本学位论文。

（注：保密的学位论文在解密后适用本授权书。）

学生签名：

指导老师签名：

日期：_____年 月 日

日期：_____年 月 日

摘要

全球互联网化的大背景下，许多传统的工作都会被互联网的相关技术所代替。手工纸质签到就是这样一种情况，当前全国高校课堂上较多采用纸质签到的传统方式，往往会增加高校授课的时间成本。签到过程通常由任课教师发一张打印了全班同学的签到表，然后由同学进行手工签到。其缺点显而易见，手工签到的效率是非常低的，有时候甚至要花费掉大半节课的时间去签到，还伴随着代签等情况。为了让这种情况得到缓解，本论文拟设计与实现一个基于二维码的考勤签到系统。方便同学高效签到。本文主要描述了二维码原理和在项目中引用二维码传参，项目的数据库表设计，考勤系统的设计和实现。通过设计该系统并实例研究，希望能够改善传统的签到方式，为课堂会议等签到方式提供更多选择。

该系统设计且最终要实现的功能是这样的：用户能进行登录，教师端能够动态生成二维码，用户能够查看数据库信息，数据能够可视化，学生端扫码后能够免登录进入签到界面。具体应用场景是：学生签到/请假在上课之前，教师打开投影仪，然后用电脑进入到该系统的教师管理页面，设置签到时间，然后选择对应需要签到的班级，屏幕会投影二维码。此时学生可以拿出手机，打开手机浏览器，使用扫一扫，可以进入到扫码签到界面，进行签到、请假、查看自己出勤记录等操作。教师则能够通过该系统查询到签到、请假的情况。

该系统使用了 B/S 模式设计，整个系统使用前后端分离的开发模式，数据库则是采用了关系型数据库 MySQL，前端使用了 Vue.js 框架进行前端代码设计，同时配合 HUI-VUE 框架，以及多种其他技术加以辅佐。后台采用 Node.js 加 Express 框架对后台逻辑进行处理。学生和教师数据会事先录入进数据库，数据库里面会有学生手机号和密码等信息。当学生使用手机端扫码访问系统的时候，第一次会需要手动输入自己的手机号和密码获得系统访问的许可。当认证成功之后，学生的数据会被保存进学生手机的浏览器的本地存储空间。之后使用扫码签到的时候无需使用登录操作。学生此后进行扫描教师发布的二维码操作之后，浏览器会从本地存储里面拿取学生信息，然后直接转到学生自己的页面，然后直接进行签到或者请假操作即可。

该系统目前已经完成了基于二维码的扫码签到，实现考勤统计。已经基本实现了任务书当中的功能。虽然界面会有瑕疵，例如表单在手机端的缩放，但不影响其功能。项目经过了用例测试、性能测试。前后端分离的两个项目，其中前端使用 Webpack 自带服务器将代码部署在服务器上，而后端使用 Node 服务器将后端部署在另一个端口上。

该系统有效的解决了手工纸质签到的各种缺点，虽然当初自己设想加入的功能还有很多没完成，但是已经能够满足基本要求，能够高效的解决签到问题，本系统可以通过设置局域网 WIFI，让学生使用手机连接教师设置的局域网 WIFI 进行系统的访问，防止有些同学远程扫码签到的问题。

关键词：考勤系统 二维码 考勤签到 考勤统计

Abstract

Under the background of global internetization, many traditional jobs will be replaced by new technologies of the Internet. The traditional way of paper check-in is such a situation. At present, this way is often used in colleges and universities classes all over the country, which often increases the time cost of teaching. Usually, a teacher takes out a student-name-printed paper form for the whole class, and then the students sign-in by hand. Its shortcomings are obvious. The efficiency of manual check-in is very low. Sometimes it even takes most of the class time to check-in, accompanied by surrogate sign-in and so on. In order to handle this situation, this paper intends to design and implement a two-dimensional code-based attendance check-in system. It is convenient for teachers and students to sign in efficiently. This paper mainly describes the principle of two-dimensional code and the use of two-dimensional code in the project, the database table design of the project, the design of attendance system and the statistics of attendance system. Through the design of the system and case study, we hope to improve the traditional check-in way and provide more choices for classroom, meetings some other efficient ways.

The functions of the system are as follows: users can sign in, teachers can publish two-dimensional codes, and they can view database information, data can be visualized, students can scan code by phone browser. The specific application scenario is: before the students check in/leave for class, the teacher opens the system, then sign in and set some information, and then chooses one class to check in, the screen will publish the two-dimensional code. At this time, students can take out their mobile phones, open the mobile browser, use the sweep, can enter the system to handle check-in or leave, or check their attendance records and so on. Teachers can see every information from database.

The system uses B/S mode design. The whole system is front-end and back-end separation. The database uses MySQL. The front-end uses Vue.js framework. At the same time, it cooperates with HUI-VUE framework and a variety of other technologies to assist. The back-end uses Node.js and Express framework to process the back-end logic. The data of students and teachers will be entered into the database beforehand. Just like some information such as students' mobile phone number and password. When students use the mobile phone to access the system, they need to enter their mobile

phone number and password manually for the first time to obtain the system access permission. When the authentication is successful, students' data will be stored in the browser's local storage. There is no need to use login operation when using next times. After the students scanned the two-dimensional code published by the teachers, the browser would take out the students' information from the local storage, then go directly to the students' own pages.

At present, the system has completed sweeping code check-in based on two-dimensional code and realized attendance. The task has been basically realized. Although the UI may have shortcoming, such as when student use phone, the table's style does not look well. The project has passed the use case test and performance test. And the front-end uses webpack's own server to export, while the back-end uses Node server to deploy the API on another port.

This system effectively solves all kinds of shortcomings of manual paper check-in. Although there are still many incomplete functions that I imagined to join at the beginning, it can meet the basic requirements and solve the check-in problem efficiently. This system can enable students to use mobile phones to connect with the local area network WIFI set by teachers to access the system. Preventing Students from Scanning Codes remotely.

【Key words】 Attendance System QR code Signature Attendance statistics

目录

第 1 章 引言	1
1.1 研究背景.....	1
1.2 研究目的和意义	1
1.3 研究动态	2
1.4 本文的组织结构	2
第 2 章 相关技术	5
2.1 二维码相关技术介绍	5
2.1.1 二维码的发明	5
2.1.2 QR Code	5
2.2 全栈技术简介	12
2.2.1 前后端分离模式	12
2.2.2 前端框架 Vue.js	15
2.2.3 后端 Node.js 简述	16
2.2.4 其他技术介绍	16
2.3 本章小结	18
第 3 章 考勤签到系统功能设计	19
3.1 系统分析与设计	19
3.1.1 需求分析	19
3.1.2 概要设计	20
3.1.3 详细设计	28
3.2 本章小结	36
第 4 章 考勤签到系统功能实现	37
4.1 系统实现	37
4.1.1 系统前端实现	37
4.1.2 系统后台实现	40
4.1.3 系统二维码生成以及移动端扫码签到代码实现	42
4.1.4 跨域	44

4.2 本章小结	45
第 5 章 系统测试	47
5.1 测试以及结果	47
5.1.1 登录模块测试	47
5.1.2 扫码签到模块测试	48
5.1.3 请假模块测试	48
5.1.4 性能测试	49
5.1.5 测试结果	49
5.2 项目部署与界面 UI	49
5.2.1 项目部署	49
5.2.2 系统界面展示	50
5.3 本章小结	55
第 6 章 总结与展望	57
6.1 总结	57
6.2 展望	57
结论	59
致谢	61
参考文献	63
附录 英文翻译	65
一、 英文原文	65
Research on the application of HTML5 in the development of mobile internet	65
二、 英文翻译	68
HTML5 在移动端的应用	68
附录 源程序	71
后台 SqlMap 部分	71
后台签到实现部分	72
前端发布二维码部分	73
移动端签到实现	74

第 1 章 引言

1.1 研究背景

互联网给人们的生活带来了便利，作为互联网的窗口：二维码是互联网的主要推力。商务推广、小程序、收付款，处处都是二维码的身影。无疑二维码深深影响了 21 世纪世界人民的生活，与全球互联网化的大环境背景下，“互联网+”模式给社会带来了诸多福利和方便。但是生活中还有很多小细节，我们忽略了它的现代化进程。例如当前的很多大型会议，还是依靠手动签到，还有很多课堂还依靠着纸质签到。手工签到的效率是非常低的，通常会花费整个会议、一整节课堂的时间，无疑降低了课堂的效率，且并不环保，工作量大，通常还会伴随有代签等问题。所以设计与实现一款能够解决上述这些问题的软件的也就成为了我选题的方向。

1.2 研究目的和意义

本论文旨在设计和实现一个基于二维码的考勤签到系统，拟解决传统签到方式的各种不方便的问题，同时为高校的课堂、会议等签到方式提供更多选择。借此提倡环保绿色生活，减少纸张和墨水的使用，有效的保护自然环境，为社会、自然、地球的未来贡献一份我力所能及的力量，为当代快节奏时代生活提供高效的小工具。

要想设计与实现该系统，需要了解二维码的原理，与利用二维码传参，项目的数据库表设计，考勤系统设计。技术可行性上，我拥有一定的技术积淀，所以项目实现起来不是什么难题。我使用了当前非常流行的语言 JavaScript 对项目进行了编码，整个项目使用的技术栈有 Vue.js、Element-UI、H-UI、Axios、Webpack、Vue-Cli3、ECharts.js、Node.js、Express、MySQL。

1.3 研究动态

二维码技术已经是相当的成熟，且随着技术的日新月异，各式各样的工具在互联网上唾手可得。国外主流签到系统较为先进，绝大多部分的签到系统采用的是指纹、面部、虹膜、掌纹等签到方式，这些签到往往十分高效，在参加会议人员进入会议室之前就能完成签到，但是成本较高。国内也有较多类似的签到系统，同时也拥有较多基于微信公众号、微信小程序的二维码扫码签到系统。对于我们即将参与设计与实现的项目，指纹、面部等签到实现方式耗费成本略高，且我们不具有该能力与技术基础。同时，基于微信平台的二维码扫码签到系统又过多，所以不适合我们再去反复造轮子。

综上所述，国内外的签到系统已经是非常成熟了，且特别方便，采用非二维码扫码签到的智能签到系统更加高效的解决了传统纸质签到的各种不足。但是这些签到平台往往较为昂贵，且国内的签到系统往往需要用户去注册使用，或者付费使用。为了制造出更为廉价使用的系统，且力图使用新鲜技术。所以我选择了当前很少应用于扫码签到系统的前后端分离技术对项目进行设计和开发。作为计算机软件相关专业的大学生，应有能力自主编写有利于身边同学的签到软件，提升高校签到效率，以更为环保创新的方式解决传统签到带来的种种不便利因素。

1.4 本文的组织结构

本文主要按照以下章节开展论述：

第一章，引言。介绍该研究课题的相关背景、研究目的和意义，以及大概的研究内容。

第二章，相关技术。介绍在设计和开发基于二维码的考勤签到系统过程中所使用的技术和工具。

第三章，考勤系统功能设计。对该系统进行需求分析、概要设计、详细设计。

第四章，考勤系统的功能实现。主要是介绍了系统前端、系统后台、关键功能的实现方法。

第五章，系统测试。对系统的功能进行整体上的测试。

第六章，总结与展望。对毕业设计进行总结和未来展望。

结论、致谢，结束语。总结在分析设计与实现部分的主要工作，并介绍后续的相关工作。

第 2 章 相关技术

2.1 二维码相关技术介绍

2.1.1 二维码的发明

二维码 (2-dimensional bar code), 它是一组于二维方向(平面)上依照一定的规律, 以黑白相间的类似像素点似得图形记录各式各样的数据符号, 然后将这些数据型号排列成特定的几何图形, 它的前身是条形码。在我们日常生活中经常遇到的二维码种类为 QR Code (Quick Response)。QR Code 被广泛应用于生活之中, 例如支付宝、微信的收付款二维码, 网址链接等。它是 21 世纪移动设备上最流行的编码方式。

QR Code 也叫做快速反应码, 它的前世是来自日本的 Denso Wave 公司(日本电装株式会社 Denso Corporation 的子公司)的一名员工: 腾弘原 (Masahiro Hara) 发明的。当时公司为了希望能通过一维码获得更多产品信息, 但是无论怎么处理, 条形一维码都很难存储更多信息。腾弘原带领团队进行了两年多的研究, 将一个又一个小方块组合在一个平面上, 形成了二维码, 这个二维码可以存储 5000 个日文字符, 是传统的条形码信息容量的 250 倍。但是腾弘原并不在乎自己发明的这个东西, 还说了一句话: “这种技术其实随便找个网络工具就能实现, 所以这么简单的东西, 我就不收专利费啦”。随后在 2011 年中国的凌空网创始人徐蔚申请注册了“二维码扫一扫的专利”。与此同时, 每年在中国, 人们通过各式各样的方式使用二维码的次数超过了 5000 亿次, 毫无疑问, 二维码是当代影响且改变了人类生活方式的编码。马化腾曾经说过“二维码会是互联网的入口。”, 如今也得以实现。

2.1.2 QR Code

在许多种类的二维条码中, 常用的码制有: Data Matrix, Maxi Code, Aztec, QR Code, Vericode, PDF417, Ultracode, Code 49, Code 16K 等。而我们最常用, 且最流

行的编码就是 QRcode，下面，我们将比较一下 QR 码和传统条形码的差异来凸显 QRCode 的优越性。

2.1.2.1 QR 码和传统一维条形码对比

传统二维码虽然是很老旧的产物，但是它也广泛应用于我们的生活当中，例如常见的商品标签、考试的考生个人信息贴条、防伪标志等等，二维码是在条形码的基础上进行的发展，所以说，传统一维条形码是现代二维码的鼻祖。条形码和二维码如下图 2.1 所示：



图 2.1 条形码到 QR 码

1) QR 码比传统一维条形码能存储更多数据

QR 二维码能够存储数百倍于传统条形码的存储量，也就是说，当一个传统条形码能处理 30 位的数据量的时候，QR 码可以存储 7089 字（仅仅使用阿拉伯数字的时候）的庞大信息量。

2) QR 码是矢量图

QR 码能够缩放，得益于方块点的矢量图设计。我们在一个平面上设立一个坐标原点 O，有两条坐标轴呈 90° 穿过这个坐标原点，分别是一条 x 轴、一条 y 轴。那么一维条形码就是沿 x 轴横向处理数据，由密密麻麻粗细不均的细线组成，它在图像缩小的时候很容易发生重叠、模糊，从而导致数据失真；而 QR 码是沿着 x 轴和 y 轴，同时通过纵向与横向两个轴方向存储信息，加上双色方块设计，很难产生模糊重叠。一维码条形码和 QR 码的缩放如下图 2.2、图 2.3 所示：



图 2.2 一维条形码缩放



图 2.3 QR 码缩放

用控制变量法来实验数据，当数据大小相同，通过实验得出，在同一个平面上，要存储这些数据，绘制传统一维码所需要的面积是绘制 QR 码所需要的面积的十余倍。传统一维条形码与 QR 码所占空间大小如图 2.4 所示：



图 2.4 传统一维条形码与 QR 码的所占空间大小

3) 360° 无死角扫码

QR 码内置定位图案，三个“回”字样的图案就是 QR 码的定位图案，用扫描装置对 QR 码进行扫描的时候会快速锁定三个定位装置，进而对二维码进行快速解码。而传统一维条形码无法做到这么方便，有且只能使用扫码工具准确对准条形码的竖条进行解码。QR 码上的定位图案标志如下图 2.5 所示：



图 2.5 QR 码上的定位图案

4) 容错能力

QR 码容错能力非常强的，条形码不具备容错率，但是 QR 码却拥有非常强大的纠错能力。QR 的纠错分为四个级别，其编码尺寸会随着其纠错等级级别的提升而增加，会占据更多的绘制空间，同时数据量也会增加。QR 码容错级别分为 L(low)、M(middle)、Q (quartered)、H (high) 四个级别，它们分别可以恢复 7%、15%、25%、30%的码字，正是因为纠错能力很高，造就了各种各样的具有艺术特色的二维码。QR 码采用的纠错算法是[1]Reed-Solomon 算法，下表 1.1 展示了其容错等级：

表 2.1 QR 码纠错能力

级别 L	约 7%
级别 M	约 15%
级别 Q	约 25%
级别 H	约 30%

5) 支持零散数据的分割与合并

QR 码因为具有 Structure Append 特点，一个 QR 码可以被打散，分割成几个小的 QR 码，这些小二维码内存储的数据是大 QR 码的子集。与此同时，这些小的 QR 码也能够被组合起来，形成一个密度更高的二维码。而传统一维条形码没有分解数据的能力。QR 码的可扩展性如下图 2.6 所示：

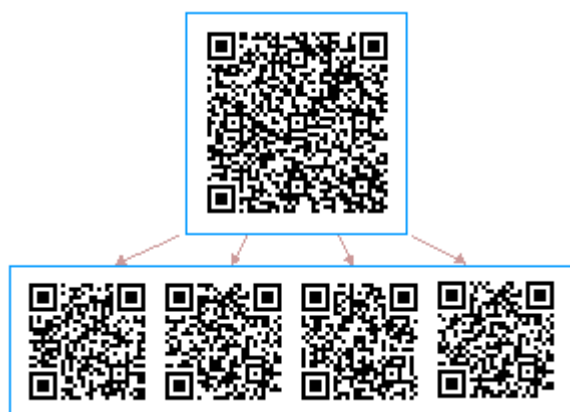


图 2.6 QR 码的可扩展性

6) QR 码能有效存储各式各样的语言。

QR 码相较于传统一维条形码，能够存储的数据格式也是各式各样，因为 QR 码的创造者是日本人，所以 QR 码是与中文汉字与日文字母相性最好的二维码编码。同等情况下，QR 码存储汉字和日文字母需要的空间更小。例如日本 KANJI 一个字母是用两个字节去表示它，最终将这两个字节码转换成以 13 字节表示的二进制码，这就是 QR 码对日文字母的编码方式；另一方面，QR 码在转换中文汉字的时候与日文汉字是类似的，所以，QR 码是非常符合我国国情的、能够被广泛应用的编码方法。

2.1.2.2 QR 码和传统一维条形码对比

QR 码虽然一眼看上去就好像仅仅是一群黑白方块拼接组合在一起的图形，但是其中隐藏着大量的信息。它就像一种语言，一种文字符号。抛开数据不谈，我们对 QR 码进行解剖：这是一种符号，它由功能图形和版本编码控制组成。QR 码的基本结构如图 2.7 所示。

1) QR 码能有效存储各式各样的语言。

(1) 空白区

乍一眼看上去就是二维码边缘的白边，看上去毫无用处，其实是为了便于二维码的识别将 QR 码与外界隔开，加大数据解码精度而存在的。

(2) 位置探测图形

三个像“回”字一样的图形，由三个分布在连续三个角落的大正方形镶嵌而成，它们主要功能是帮助扫码工具能 360° 对二维码进行扫描。主要功能是定位。

(3) 位置探测图形分隔符

它们主要是将位置探测图形与数据和定位图形分割开来，便于区分。

(4) 定位图形

由两条线段交叉组成，它们是由黑白相间的方块组合形成的线段，其主要功能是配合位置探测图形，在扫码工具对 QR 码进行扫码的时候形成两道辅助线，校正二维码的数据解码的读取角度。

(5) 校正图形

类似“回字形”，由黑正方形嵌套一个白正方形然后最内层再嵌套一个黑正方形，它们均匀的分布在 QR 码的数据区内，方便解码工具在扫描 QR 码的时候确定二维码的数据数据和中心，防止数据读取错误。

2) 编码区

(1) 格式信息

展示 QR 码的纠错级别，纠错级别总共 4 个等级，分别为 L、M、Q、H 四个级别的纠错级别。

(2) 版本信息

记录 QR 码的版本信息，QR 码有不同版本多达 40 种不同规模的矩阵，规模越大则模块越多，最小为 21×21 （版本 1），最大为 177×177 （版本 40）。

3) 数据和纠错码

保存了用于修正数据的字符、数据信息。当且仅当数据信息在 QR 码设定的容错等级范围内，纠错码可以对破损数据进行修正，使得解码正确进行。

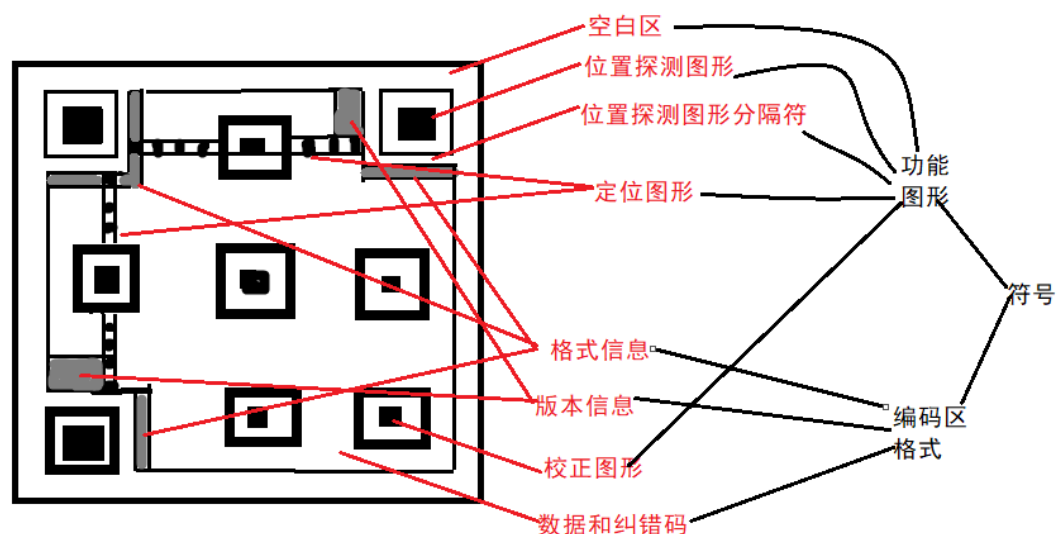


图 2.7 QR 码的基本结构

2.1.2.3 QR 码的编码

QR 编码方式是将数据以两个字符划分为一段，然后对这段当中的第一个字符乘以 45，然后再和第二个字符相加。如果剩余一个字符，那么直接用 6bit 表示。如果是 8bit 字节数据，那么将不会经过编码，而是直接被保存。

2.1.2.4 QR 码的分类

QR 码有三种类型，它们分别是：

1) QR 码:模型 1/模型 2

- ① 模型 1 是 QR 码最早期的产品，它的旗舰版本是 73x73 码元。
- ② 模型 2 的旗舰版本是 177x177 码元，也就是我们日常生活中最常见到的二维码，它是由模型 1 改进而来的。相较于模型 1，模型 2 可以比模型 1 多处理 6000 左右位数字，因为其存储量与方便性，成为了当前互联网产业的必备品。

2) Micro QR 码

Micro QR 码最显著的特点就是它只有一个定位图形，最高可以存储 35 位字母数字，最高是 17x17 码元。

3) IQR 码

IQR 码可塑性非常强，它可以天花乱坠的修改。它支持黑白反色、圈点图形、里外翻转。可存储和处理 40000 位数字，其旗舰版本是 422x422 码元。



图 2.8 QR、Micro、IQR 码(从左到右)

对于项目中要是用的二维码就是 QR 码的模型 2。如今支持生成 QR 码的 JavaScript 库挺多，从 qrcode.js 到 qr-image，大部分二维码都是用 Python 语言编写的，本次项目引用了 Qrcode，接下来将描述 Qrcode 的功能与接口。

2.2 全栈技术简介

2.2.1 前后端分离模式

以前，我们接触最多的就是 JSP+Servlet 的 MVC 开发模式，后来有了 Spring+Struts+Hibernate 的三大框架的 JAVA 时代，再到现在非常流行的以前端框架为主的 MV*时代,时代的进步与技术的迭代让现在的软件系统更加的灵活多变。而前后端分离的开发模式正是诞生于现在前端框架绽放的时代。在工作当中，前后端工作人员在双方规定好接口之后，可以完全的分开形成两个独立的工作站。作为前端人员，他们仅仅需要的是关注于页面的样式和动态数据的解析以及渲染。而作为后端人员，他们则更加的专注于后台的具体业务逻辑和数据处理。前后端分离的项目，以前后端为界，分别放在不同的服务器上，分别独立部署。他们具有如下的优点：

1) SPA(single page web application)单页面应用的性能

以前最流行的框架就是 JQuery.js，以直接操作一张网页的节点元素的多页面 Web 应用程序，所以网页速度和性能不会太高。自从 Ajax 流行起来之后，Web 应

用程序开始变得复杂了起来，所有操作都是在一张页面上完成的，且数据渲染和数据获取都是由前端本地完成，所以一张网页的渲染速度将会更快，页面跳转不会再刷新页面，而是在 URL 中使用#号进行视图地址的改变，页面改动速度更快，你看到的改动其实就是这张页面上的一个组件被替换了而已。也就是说所有数据都是通过 Ajax 交互的。

2) 解决前后端开发职责不清的问题

以前，前后端开发十分受限制，因为后台需要前端把 HTML 页面做好之后，拿过来转换成他们服务器所需的模板引擎，最经典的模板引擎就是 JSP。前端往往受限制于后端，无法做更多事情，后端工作量也很大，处理数据交互等问题。有时候前端处理数据交互，往往需要自己书写模板引擎与一些后台逻辑，对前后端的职责划线不清。

3) 解决开发效率的问题，前后端不用再相互等待

以前，前后端工作人员一起开会，他们会首先听取项目经理给他们提的需求，然后就开始工作了，后端工作人员会坐在办公室喝茶，等待前端工作人员把图切出来、实现前端页面之后，后端工作人员才会开始工作。然后前端工作人员会坐在办公室喝茶，等待后端工作人员的需求改变和界面数据渲染 BUG。且非常致命的是，作为公司的测试人员，他们从一开始就喝茶，喝到前后端工作者好不容易把项目开发完，然后才能开始测试。一旦发现 BUG 就会重复以上动作，这对于公司的项目交付影响是非常大的。

现在，前后端开发人员、测试人员需要在一起开会。听了需求之后，制定一个前后端分离的 API 开发文档，三方就可以开始着手开始工作，无需互相等待，前后端一起开展工作，测试可以随意参与前、后端任意一方的测试，减少后期的工作量。下图 2.9 和 2.10 分别展示了前后端未分离和分离开发的项目流程，可以看到图 2.9 中前端开发和后端开发在一条线上，会互相等待，而在图 2.10 中，前端后端完全分离，双方只需要各自通过测试即可。

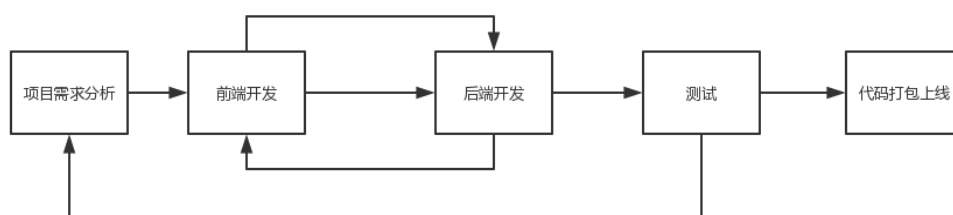


图 2.9 前后端未分离开发的项目流程

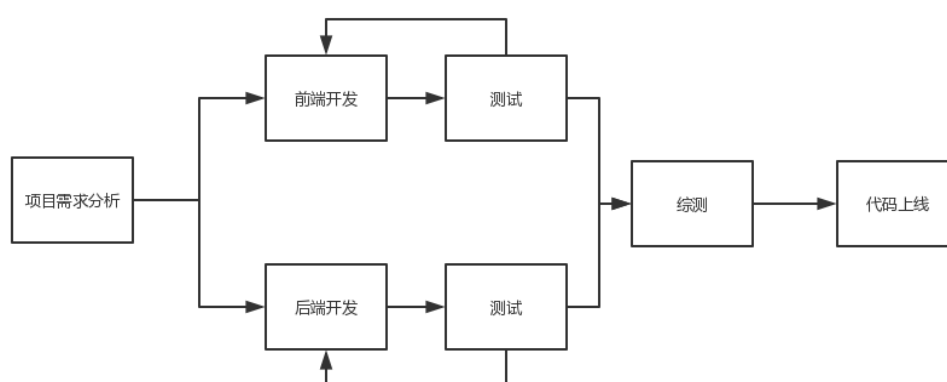


图 2.10 前后端分离开发的项目流程

4) 减少代码耦合度

代码耦合度高是在使用模板引擎的时代，往往可以看到脚本，服务器语言全部糅合在一个前端页面文件当中，这对运营和维护非常不利。在前后端分离之后，前端部分只有前端逻辑，后端部分只有后端逻辑，减少了依赖，利于项目的后期调试。

同时他们也有如下缺点：

1) 前后端学习成本增加

可想而知，在前后端分离开发之后，前后端的工作流程将会发生翻天覆地的变化，之前所学的模板引擎等全部不再适用，后端将花更多心思去优化数据端服务器，提升算法质量，而前端需要写更多脚本代码，处理多终端的样式，以及考虑数据渲染问题。

2) 数据交互过度依赖接口文档

数据交互过度依赖接口文档，当前端开发完后，发现和后端 API 对不上会加大工作量。当 API 格式出错，双方将会出现更多的难以处理的 BUG 和工作。

3) 前端工作量加大

无论后端采用 NodeJS，还是 JAVA 等语言，前端都会多加一层 NodeJS，因为作为数据交互，前端需要控制 Controller 层。导致前端工作量加大，更难维护前端的部分，但是可以减少测试的成本。

4) SEO 的难度加大

因为是单页面，无法像多页面 Web 应用程序那样，做到每一个页面的关键词暴露，不利于提高除开主页之外的其他页面的 SEO，这部分会加大后端工作人员的难度，这个工作由后端人员承担。

5) 后端开发模式的迁移增加成本

后端工作人员由于操作不了前端，在前端不变的情况下，更新后台 API 和迁移开发模式会更加困难。

2.2.2 前端框架 Vue.js

MVVM 是前端的一种开发思想，它将前端视图层进行了分层开发。以当前火热的的前端三大框架(Angular.js、React.js、Vue.js)之一 Vue.js 为例，因为 Vue 所构建的应用是单页面应用程序，它也是一种 MVVM 框架：

- ① 其中 M 就是这个 Web 应用程序每一个组件里面的单独的数据。
- ② 其中 V 就是这个 Web 应用程序的视图层，相当于 HTML+CSS 结构的部分
- ③ 其中 VM 就是作为 VM 的调度者，作为 V 和 M 之间的“桥梁”，为数据提供了双向数据绑定，动态改变页面的数据而不去刷新页面，组件渲染使得 Vue 应用程序性能优异。

那么 Vue.js 到底是什么呢，我们曾经很爱使用 JQuery.js 来频繁操作 Web 应用程序的 DOM 来更新页面，但现在，我们不必再去操作 DOM 了，Vue.js 就是这么一款解放 DOM 操作的框架。一个基于 Vue.js 的 WebAPP 是由一个一个组件堆叠

起来的，一个页面可以看成多个组件组成的。它主要的特点是数据驱动模型。

传统的数据改变需要手动改变 DOM 来改变页面中的视图，而 Vue.js 仅仅需要对数据进行操作就能动态改变视图。不仅如此，三大框架还能让移动端的 WebAPP 更加逼近原生的 APP 的体验。

2.2.3 后端 Node.js 简述

Node 是一个 JavaScript 在服务端运行的开发平台，成功的让 JavaScript 成为一门服务端语言。Node.js 是一个基于 Chrome V8 引擎的 JavaScript 运行环境。Node.js 使用了一个事件驱动、非阻塞 I/O 的模型，使其轻量又高效，还有配套的包管理工具。具有良好的生态环境，非常适合前端人员入门的一个后端平台。

2.2.4 其他技术介绍

1) ECharts

ECharts 是 web 前端接触大数据的大门，是百度出品的一个数据可视化的 Js 库，它可以做任何形式的表，不管是抽象的还是正式的。只需要定义好数据格式与图表的容器，就可以通过使用 Ajax 异步抓取后台数据，动态更新图表内容。在本项目中，我们需要使用 Axios 异步获取后台数据，然后将数据通过 Vue.js 的双向数据绑定的方式渲染数据，然后用 Canvas 绘制图表，本项目只需要考虑如何让 ECharts 和 Vue 进行兼容。可以说 ECharts 赋予了用户对于数据进行挖掘整合的能力。与此同时在拥有众多交互特性下 ECharts 依然可以做到直角系图表（折、柱、散点、K 线）20 万数据秒级出图。

2) Axios

Axios 就是一个基于 promise 的 Http 库，就像我们常使用的 Ajax，不过添加了 ES2015 的一些东西，并提升了安全性和便捷性。其特点有：

- (1) 于浏览器中创建 XMLHttpRequests
- (2) 从 node.js 创建 http 请求
- (3) 支持 Promise API

- (4) 拦截请求和响应
- (5) 转换请求数据和响应数据
- (6) 取消请求
- (7) 自动转换 JSON 数据
- (8) 客户端支持防御 XSRF

3) WebStorage

WebStorage 是 HTML5 带来的 API，是 Web 应用程序的本地存储解决方案之一，它解决了 cookie 的不安全、存储容量小的问题，同时又具有 cookie 的所有功能，WebStorage 分为：

①localStorage(本地存储)，顾名思义，这个 API 帮助我们将信息存在本地，当浏览器关闭的时候仍然会保存在硬盘上，可以用于长久存储，保存用户名、密码等。

②sessionStorage(会话存储)，就是在浏览器窗口持续期间，与设置了会话存储的网站进行了会话，则在浏览器关闭之前，这些数据会一直存在。只能用以短期存储，可以用于安全性需求较高的网站。

4) Vuex

Vuex 是一个为 Vue 单页面程序的组件之间进行通信的工具，一个为中大型单页面 WebApp 统一管理状态信息的一种模式。

在本程序中他起到了管理用户登录状态，和响应式的管理页面导航栏组件里用户信息与其他组件之间的关系联动。

首先在 store 文件夹下建立一个仓库，这个仓库用来存储用户的特征，以便在组件之中实现通信。这个仓库命名为 userStore.js。在其中，我们需要创建一个 Vuex 对象，它包含 state、mutation、action、getters 这四个属性，我们在 state 中定义要存储和共享的数据，在 mutation 中定义改变 state 中数据的方法，在 action 中我们提供对外的接口，外部想触发 mutation 中的方法必须通过 action 中定义的方法，在 getters 中，我们设置获取 state 中数据的方法。

5) Vue-router

vue-router 是 vue 中的路由管理器，因为 vue 没有多页面跳转的概念，router 就是以浏览器的 hash 栏里以#号结尾的路由（锚点）。所以 vue-router 的“页面跳转”实际上就是组件的过渡。

6) Nginx

Nginx 是一个俄罗斯开发的 http 轻量级服务器，它与 Apache 不同，它的服务器稳定，内存低，反向代理能力强。它是由俄罗斯工程师 Igor Sysoev 用 C 语言开发的高性能 Web 服务器。Nginx 开源，轻量级是本次项目选择它的原因。Nginx 可以不仅仅只作为一个 HTTP 服务器对一个 Web 项目进行发布，还可以反向代理各个服务器。

2.3 本章小结

本章主要对项目中主要使用的一些技术点进行了介绍，在对一个项目进行开发之前，必须先了解实现它的技术。只有将基础打好，才可以有能力对接下来的项目进行分析设计，并且将项目付诸于实际编码，最终实现目标。本章节的所有技术将会在开发过程中使用到，所以本章节是下一章节的铺垫。

第3章 考勤签到系统功能设计

3.1 系统分析与设计

3.1.1 需求分析

基于二维码的签到考勤系统是我们预先将同学的所有信息录入进数据库中。然后在当老师登入系统之后，老师发布二维码，同学进行扫码，直接进入到签到界面进行签到。二维码是实时生成，且内含有应签到时间、目标地址等信息。学生进行扫码可以进行签到。下面将对系统进行需求分析。

3.1.1.1 功能需求分析

该系统至少需要两个登录方式，分别为老师或学生提供登录的接口。当老师登录后能够设置签到时间、发布二维码、以及查看学生的签到信息。学生能够通过手机浏览器扫描老师发布的二维码进入签到页面，学生能够签到、查看自己的签到情况，签到后数据会被录入数据库。还有够直观展示数据的图表，这些数据通过后台传入，能够渲染页面形成图表，做到数据可视化。因为学生都是用手机或者 Pad 进行签到，所以页面应该做成响应式，兼容各个可视窗口大小，并保持页面样式风格。

为了用户的体验，界面应该无刷新，渲染速度快，界面友好。

故，本系统应该拥有如下功能点：

- ① 用户登录
- ② 二维码动态生成
- ③ 查看数据库信息
- ④ 数据可视化
- ⑤ 浏览器存储用户信息。
- ⑥ 学生签到/请假

3.1.1.2 系统配置和开发环境

本系统为了实现稳定可靠，稳定快速响应用户请求，采用了 Nginx 服务器，硬件方面：系统是 Windows10、内存 8G，用智能英特尔酷睿 i7 四核处理器，数据库采用 MySQL，开发环境为 VSCode 2019、Node.js8.0 或以上版本、Vue-cli3.x、GitBash，Nginx v1.13.0 或以上版本。

3.1.1.3 性能需求

要求响应速度得到保障，登录反馈时间不超过 3s，签到时间不超过 1s，学生扫码后能够尽快进入签到页面，且签到后的数据能够及时传输到数据库中，图表渲染在 2s，为了给用户体验，需要采用 Nginx 搞笑高性能的服务器，以解决其他服务器冗杂，缓慢的问题。

3.1.1.4 其它需求

界面友好，使用 Vue.js 构建前端页面，要求性能高效，后台代码可维护性好，能对 API 进行分类。

3.1.2 概要设计

3.1.2.1 对二维码签到系统的功能描述

本系统将由下列参与者进行参与：

表 3.1 系统的参与者与描述

参与者	描述
学生	系统的具体参与者，是用户-学生的集合
老师	系统的具体参与者，是用户-老师的集合

下面我们将分别针对老师和学生，对其用例进行分析。首先是我们的老师，老师能够通过教师通道进行登录，老师有权限设置签到时间，老师有权限查看班上所有成员的签到情况和请假情况，老师能够发布二维码，老师能够退出系统，老师能够看数据表。如下图 3.1 教师用例图所示：

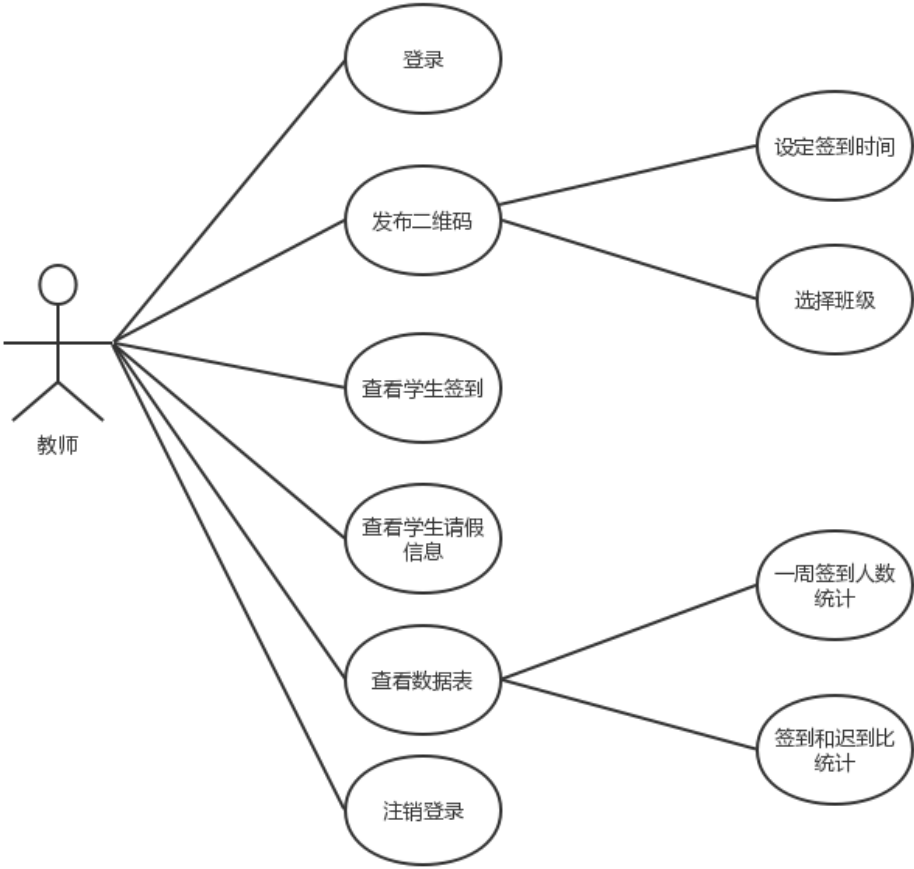


图 3.1 用户-教师用例图

下面主要对教师发布二维码这个用例进行用例实现设计，如下图 3.2 教师发布二维码用例图所示：

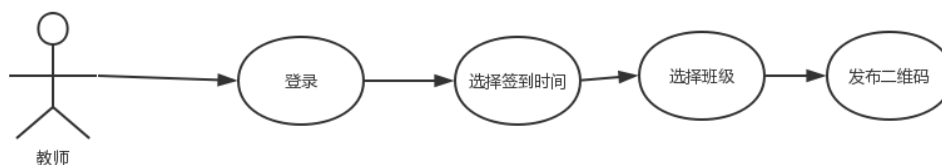


图 3.2 教师发布二维码用例实现

这个用例是教师发布二维码，用例的前提是教师通过教师登录口登录，然后进入教师管理界面，在设置中选择签到时间，或者默认下一个整时点，然后在二维码发布页选择要签到的班级。这个时候，`qr-code.js` 能够接收到时间和班级参数，立马生成二维码。这个流程需要很多细节，例如结合 `axios`，在页面进入到教师的管理页面的时候需要及时的通过 `axios` 异步动态的渲染到数据，这样才能给教师选择的机会。

分析完教师的其中一个用例实现之后，接下来我们对学生这个用户群体进行用例分析。首先，我们的学生，能够通过学生通道进行登录，学生需要扫描老师发布的二维码才能正确获取参数，才能签到成功，学生能够请假，学生能够查看个人签到信息，学生能够看到总体的数据表，学生能够注销登录。如下图 3.3 学生用例图所示

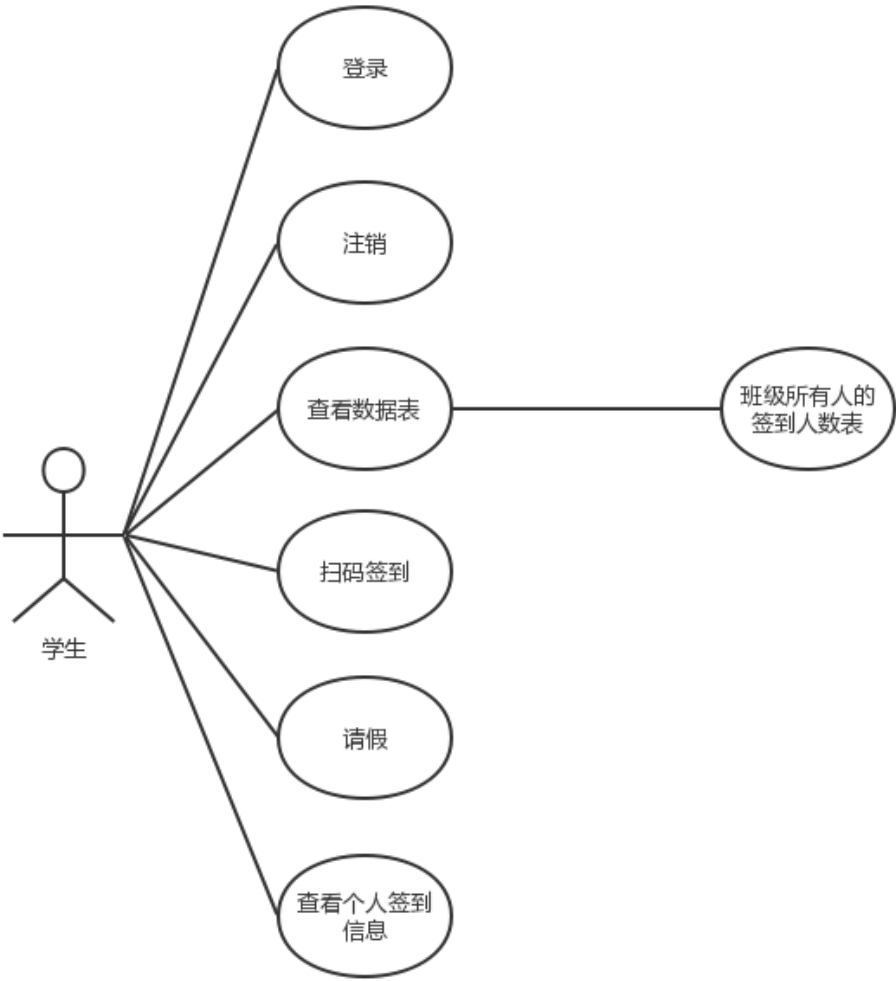


图 3.3 用户-学生用例图

然后我们主要对学生扫码签到这个用例实现进行设计：如图 3.4 学生扫码签到用例图所示：



图 3.4 用户-学生用例图

该用例是学生通过手机浏览器对教师发布的二维码进行扫码，前提是学生曾经登录过这个系统，这样手机浏览器的会对学生信息做一个本地存储（localStorage），当学生扫码后，浏览器会接收到网页的 URL 和两个参数，同时页面会自动判断本地存储中是否有用户的信息和 token 验证，有的话就直接将组件过渡到学生管理界面，学生通过点击签到按钮，前端会动态生成一个当前时间，与教师发布二维码里面的时间参数进行对比，从而决定学生是否迟到和签到成功，页面会对签到行为进行反馈。

下面我们对老师和学生两类用户的流程图进行详细设计，针对教师的操作，我们设计了如下的流程图，如图 3.5 教师流程图所示。

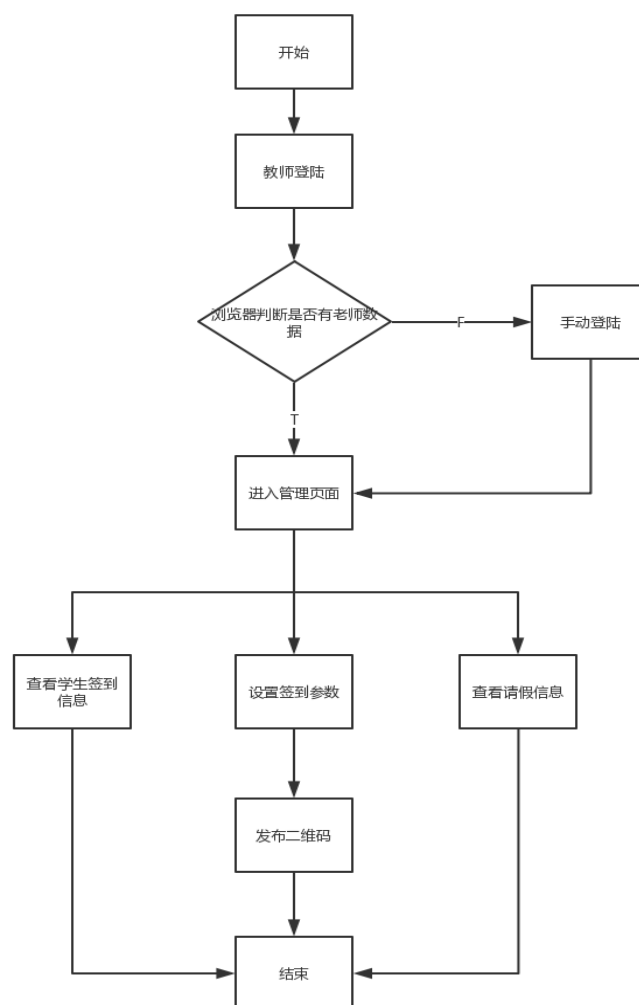


图 3.5 用户-教师流程图

然后对学生的操作流程设计如下图 3.6 所示：

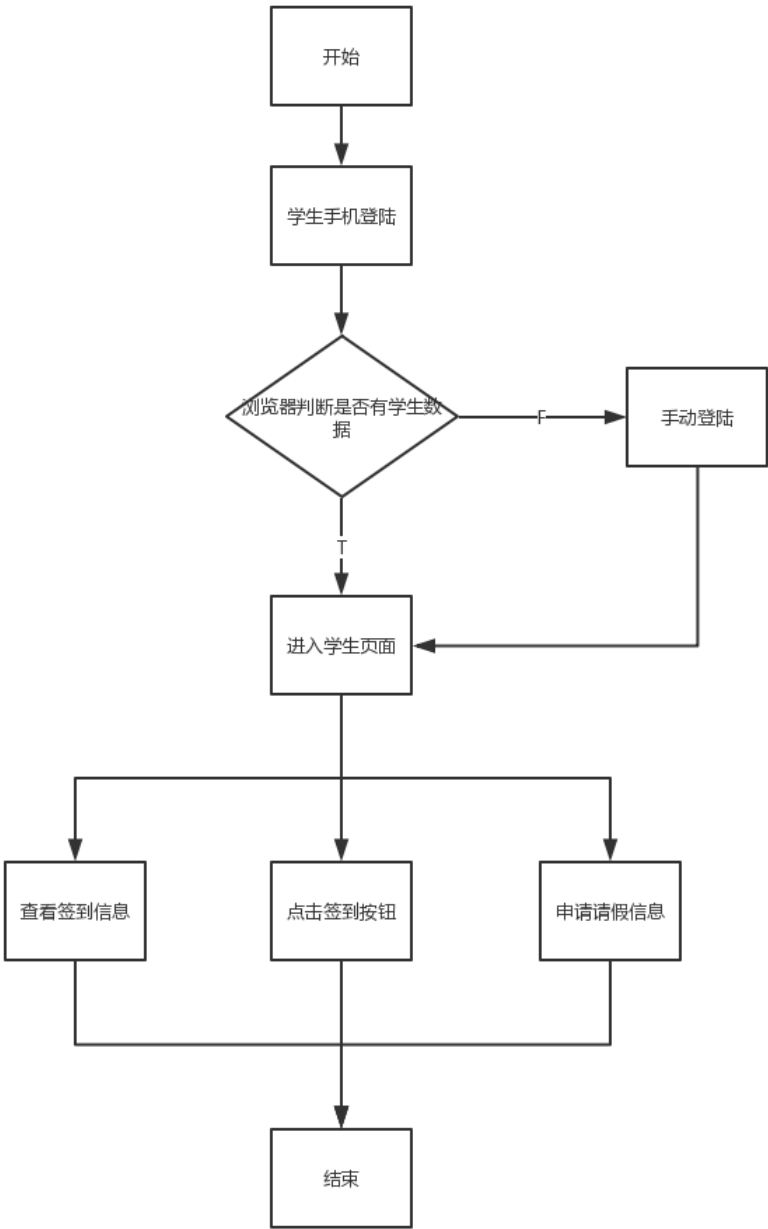


图 3.6 用户-学生流程图

用户用例和用户流程图设计完了将十分有利于我们的编码工作，为我们的编码工作绘制了蓝图，接下来我们只需要认真的分析系统应该采用那种结构。我们将系统结构设计好后还需要规划系统的 UI，这样就能将我们的项目稳步进行。

3.1.2.2 系统结构

系统总体结构任然是 MVC (model, view, controller) 模式，尽管是前后端分离开发，前端采用了 MVVM 框架的情况下，因为采用 MVC 三层结构来设计系统会让系统变得更容易维护，系统划分层次会让我们的编码效率变高。但是这又不纯粹的是 MVC 模式，因为前端控制了 V，还控制了部分 M 和 C。

在我们的系统结构中，我们将项目分为三层，如下图 3.7 系统架构图所示：

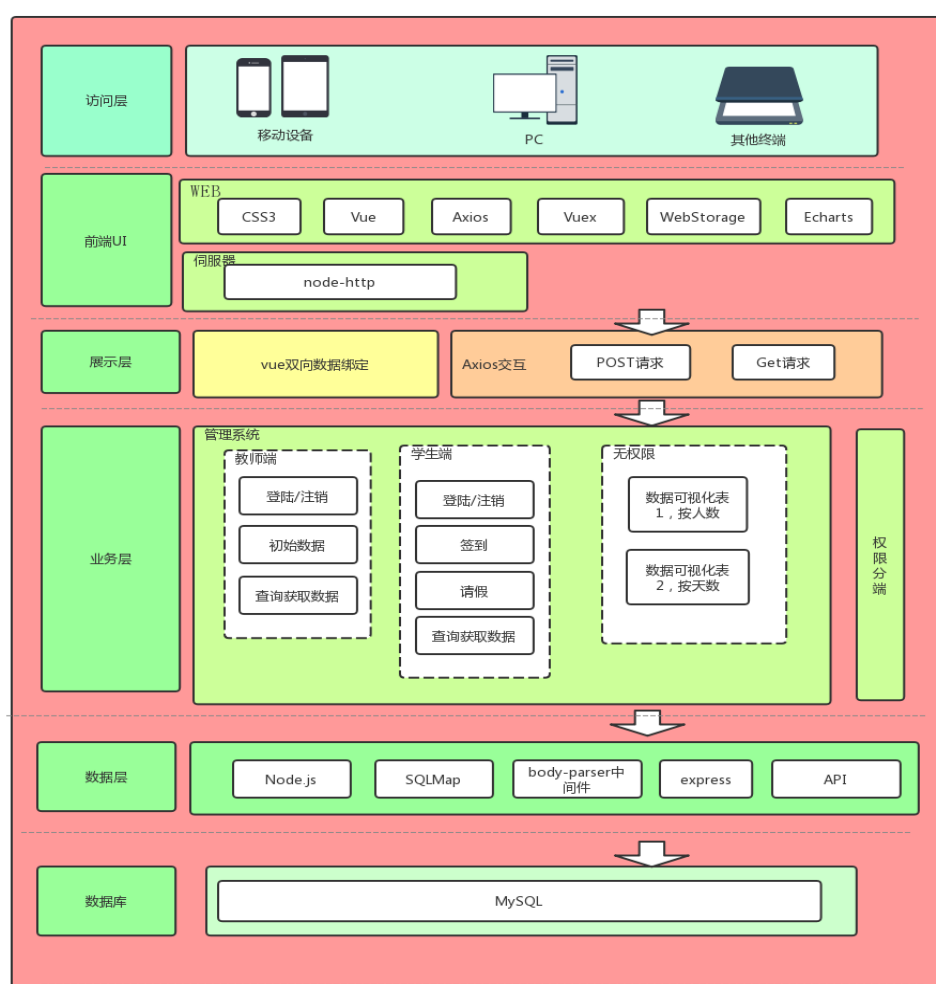


图 3.7 系统架构图

1) 界面 UI 展示层

UI 层，我们使用 VueCli3.0 脚手架对项目进行搭建，以用户进行分类就

是两个独立的界面，一个是教师端，也就是 PC 端，一个是学生端，也就是手机端。界面 UI 上的数据渲染和交互有部分是在浏览器上的缓存，还有大部分是通过后台获取。前端服务器在开发的时候直接是 VueCli 内置的 HttpServer 插件。页面组件过渡的时候需要使用 Animate.css，其基础是基于 CSS3 的，让组件过渡更加流畅。展示层主的数据交互主要通过 Axios 异步发起 Post 或者 Get 请求，请求服务器端的数据，然后通过 Vue 的双向数据绑定达到动态更新前端界面 UI 的功能。然后 Echarts 主要是为了直观展示签到数据，数据也是通过 Axios 请求后端数据，然后将数据作为参数赋值给一个 Echarts 对象。最后，为了加强各个组件之间的通信，使用 Vuex，为了让浏览器端能够保存一些关心数据和保持数据长久化，应结合使用 Vuex 和 WebStorage。这样前端页面基本就能完成了，但是由于学生扫码签到可能是手机端或者是 ipad，我们需要将整个项目做成响应式。所以为了让系统兼容各个端，且保持风格不变，我们使用了 HUI（一款基于 Vue2.0 的 UI 组件），它能够在移动端有较好的表现。

2) 业务逻辑层

业务逻辑层前后端设计都有参与，在后端部分，我们需要接收前端传来的数据，通过 body-parser 将数据从 request 请求中解析出来，然后通过 Node.js 引入的 MySQL 中间件结合自己编写的 SQL 操作数据，然后根据数据库操作的反馈结果，以指定的端口暴露 API 然后向前端请求反馈响应码。前端拿到数据之后，把这些数据进行处理。对于 MySQL 的时间处理需要得到特别重视，因为在 MySQL 中，设置数据格式为 datetime 之后，时间数据格式是：“2019-04-18 18: 05: 02”，这时候需要前后端在存储渲染数据的时候将时间转换格式。

当用户登录成功之后，后端需要给前端反馈一个 token，表示登录成功的 token，前端需要把这个 token 存入进 WebStorage 和 Vuex 中，用以控制、操作组件之间的通信。

3) 数据层

数据层主要是做数据持久化，主要是设计好用户对数据库的增删改查操作，和数据库的链接等。

3.1.3 详细设计

3.1.3.1 系统环境配置

Windows10 上，基于 Node.js 平台 8.0.0 版本以上，通过 npm 包管理工具安装 vue-cli3.x，通过 bash 命令行键入：

```
Npm install -g @vue/cli;
```

```
Vue create project
```

我们通过 vue 的脚手架搭建了 vue 的环境，然后我们需要引入项目中未来编写代码，实现某些功能所需要用到的依赖项目。在根目录的 package.json（项目中标注依赖的文件）中新建 “dependencies”键值对，并在其中写入依赖,如下所示：

```
"dependencies": {  
  "@vue/babel-helper-vue-jsx-merge-props": "^1.0.0-beta.3",//支持 jsx 语法插件  
  "animate.css": "^3.7.0",//支持 CSS3 动画的动画库  
  "axios": "^0.18.0",//比 Ajax 优秀的，能发起异步请求的工具  
  "body-parser": "^1.18.3",//对前端发送过来的字段进行解析的后台中间件  
  "echarts": "^4.2.1",//百度数据可视化工具库  
  "express": "^4.16.4",//基于 Node.js 的一款开源框架  
  "formidable": "^1.2.1",//用于文件上传的第三方库  
  "hui": "^2.0.0-alpha.8",//基于 vue 的 UI 框架  
  "mysql": "^2.16.0",//基于 node.js 平台对 mysql 数据库的一种支持  
  "vue": "^2.5.2",//vue 的版本  
  "vue-router": "^3.0.1",//vue 路由插件，能够实现前端路由  
  "vuex": "^3.1.0",//组件通信的一种方法  
},
```

然后我们进入根目录使用 bash 命令行键入：

```
Npm install --save
```

这些依赖就自动的下载到了 node_modules 中了，后续如果需要使用这些工具，我们仅仅需要 import(导入)到我们需要使用这个工具的文件里。

对于后台，我们定义了一个 app.js 文件，把后端框架和 body-parser 引入，引

入 MySQL 工具，然后开启 MySQL 服务。这样后端环境也搭建好了。

3.1.3.2 二维码功能模块实现

通过项目根目录的 main.js 引入各个工具模块，经过考量和调查，我们选用了符合我技术栈的二维码库：QRcode.js。它的使用方式较为方便。仅需要通过 NPM 包管理工具键入：

```
`npm install qrcode --save`  
然后在 Vue 文件的 script 标签里键入：  
`import qrcode from 'qrcode'`  
  
// 简单方式生成二维码对象  
new QRCode(document.getElementById('qrcode'), 'your content');  
// 设置参数方式  
var qrcode = new QRCode('qrcode', {  
  text: 'your content',  
  width: 256,  
  height: 256,  
  colorDark : '#000000',  
  colorLight : '#ffffff',  
  correctLevel : QRCode.CorrectLevel.H  
});  
  
// 使用 API  
qrcode.clear();  
qrcode.makeCode('new content');
```

下面我将对 QRcode 的参数进行说明：

表 3.2 QRcode.js 参数说明

名称	默认值	说明
----	-----	----

Element	100~500km	显示二维码的元素或该元素的 ID
Option	LEO/HEO/GSO	参数配置
Width	256	图像宽度
Height	256	图像高度
TypeNumber	4	
colorDark	#000000	前景色
colorLight	#ffffff	背景色
CorrectLevel	L	容错级别
MakeCode(text)		设置二维码内容
Clear()		清除二维码

在 teacher 的管理页面中，我们需要引用 QRcode 库。然后教师通过设置要学生进入的 URL 和当前应该签到的时间还有班级编号，将这些数据变成参数，放入二维码的数据编码区，所以，代码如下：

```

createQr(cid,url,time){
  const qrcode = new QRCode(document.querySelector('#qrcode'), {
    //text: 'your content',
    width: 100,
    height: 100,
    colorDark: '#000000',
    colorLight: '#ffffff',
    correctLevel: QRCode.CorrectLevel.H, //容错级别
    useSVG: true
  })
//传入参数，URL,班级 CID,应该签到时间 time
  qrcode.makeCode(url+'?did='+cid+'&time='+time)
}

```

与此同时我们需要在 teacher.vue 的 template 标签内放入一个能够挂载二维码的 Dom 节点：

```
<div class="qr_img">
```

```
<svg id="svg" width="300" height="300">
  <g id="qrcode" ref="qrcode" />
</svg>
</div>
```

其中在中间有个标签叫 SVG，这是可缩放的矢量图形，当我们对 SVG 标签内的 QR 码进行放大和缩小的时候并不会产生锯齿。

3.1.3.3 移动端签到模块设计

学生端扫码怎么做呢，现在的浏览器能够直接解析二维码，然后会将我们的 url 和参数直接显示在浏览器上面的地址栏里，所以，我们将采用 JavaScript 中的 BOM 方法去操作浏览器的 Hash 值。

首先定义一个函数，这个函数接收一个参数，传的这个参数就是我们需要从浏览器的地址栏里面找到的那个参数，然后用正则表达式将参数匹配并且取出：

```
//获取浏览器地址栏里的 url
export function GetQueryString(name) {
  var reg = new RegExp("(^|&)" + name + "=[^&]*(&|$)");
  var r = window.location.search.substr(1).match(reg); //抓取参数，将参数获取
  if (r != null) return unescape(r[2]);
  return null;
}
```

这样我们就成功拿到数据了，就很容易完成签到这个功能了，现在仅仅需要的是将这些数据与学生签到的时候的时间进行对比，确定签到的结果，然后将这个结果存入数据库里面。

当我们扫码进入页面的时候，通过上述函数会获取到签到时间参数，然后系统登录页面会请求浏览器的本地存储数据，抓取学生数据，进行免登录进入签到页，学生点击页面上的签到按钮就会把签到时间与传递过来的时间进行对比，来确定学生的签到时间是否满足教师设置的时间。

3.1.3.4 系统显示模块设计

从程序入口开始，我们应该先跳转到登录页面，检测浏览器上有没有用户的登录特征 isLogin 和用户的 id、name 等值，没有的话就会需要用户进行的登录，有的话就会跳转到该用户对应角色的操作界面，为了让界面有个整体的编码构思，现对系统显示模块进行设计，如图 3.8 所示：

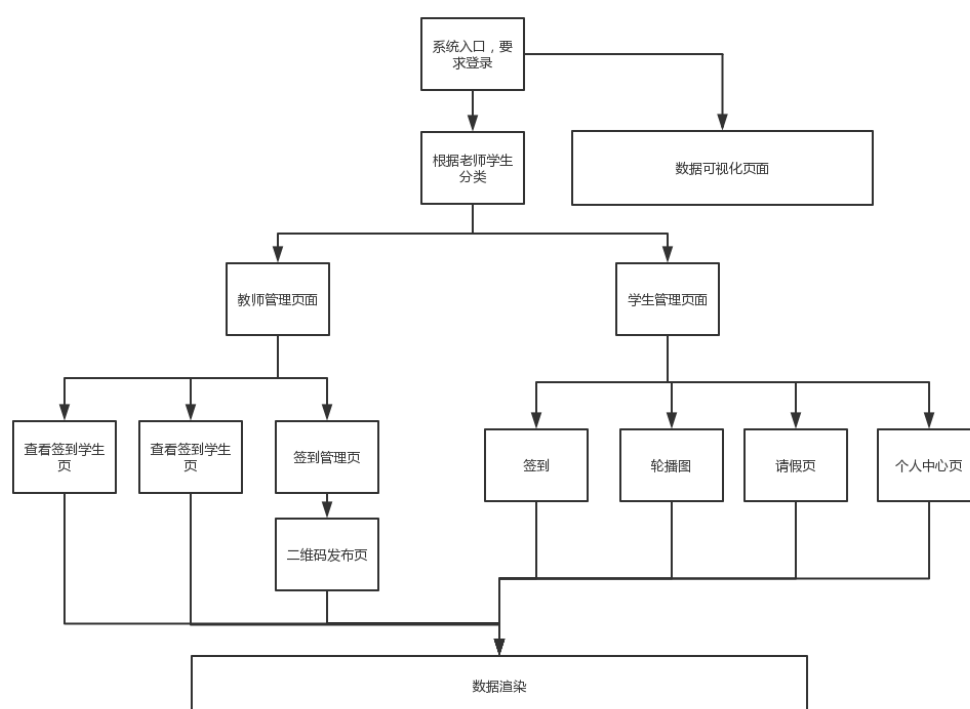


图 3.8 系统显示模块

3.1.3.5 数据库 E-R 图，表设计.

对于整个系统来说，处理好数据存储也是很重要的。例如我们如何去设计教师和学生数据表，和表之间的关系。针对这些情况，我们打算将老师和其所管理的班级，还有签到情况，以这三个点为核心去建表，然后去绘制 E-R 图，然后如果后续有其他功能，我们会在 E-R 图上进行更新，对表的设计进行完善，设计关键功能部分的 E-R 图 3.9 如下

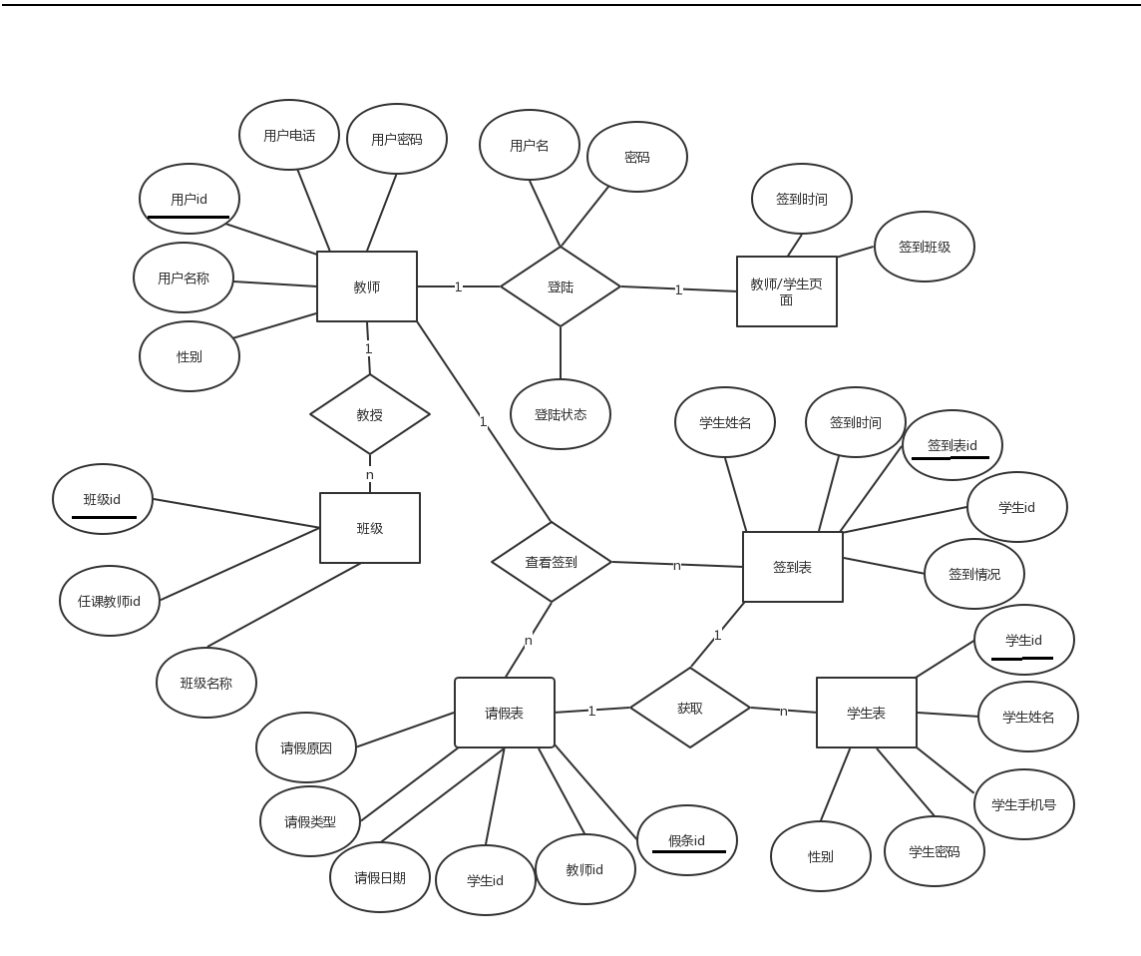


图 3.9 系统关键部分 E-R 图

接下来我们详细对系统数据库表的内容进行设计。首先，我们需要学生和教师的用户类，然后需要班级类，班级类与教师表、学生表有关联，然后还有个签到表，然后还有系表，用来存储教师管理的班级对应的系，还应有请假表。

我们先将所需要的数据罗列在如下 3.3 – 3.8 表中，对应的数据类型将在 3.10 类图中进行详细展示：

表 3.3 dictionary 数据库表

编号	Did(唯一)
学院名称	Dxyname
系名称	Dxiname
专业名称	Dzyname
年级	Dnjname

班级名称	Dbjname
------	---------

表 3.4 teacher 数据库表

编号	Tid（唯一）
电话	Tphone（唯一）
姓名	Tname
性别	Tsex
密码	Tpassword

表 3.5 student 数据库表

学号	Sno（唯一）
姓名	Sname
性别	Ssex
电话	Sphone(唯一)
密码	Spasword
班级编号	Classno

表 3.6 teacherclass 数据库表

编号	tcid(唯一)
老师编号	Tid
学院编号	Did

表 3.7 qdrecord 数据库表

编号	Qid（唯一）
学号	Sno
签到日期	Qdate
签到时间	Qtime

上课时间	Qstarttime
缺勤状态	Qteachermsg

表 3.8 qingjiarecord 数据库表

编号	Qid（唯一）
学号	Sno
请假类型	Qtype
请假原因	Qreason
请假时间	Qdate
请假人名	Sname
请假管理老师	Teacherno

对部分表设计完成之后，我们对类图作出如下 3.10 图设计：

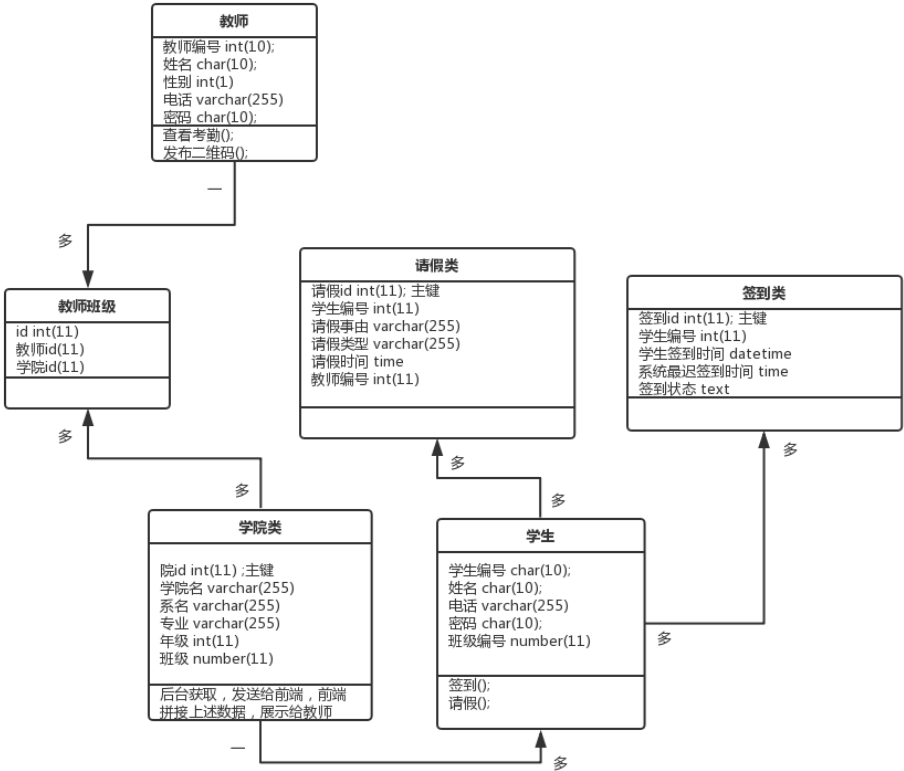


图 3.10 系统主要类图

3.2 本章小结

对于整个系统来说，我们已经为系统的实现绘制好了蓝图，本章节，我们对于系统进行了功能上的需求分析、概要设计以及详细设计。功能点上，我们主要对二维码生成以及扫码签到模块进行了设计。然后在详细设计中，我们对关键部分的功能实现进行了设计。然后我们对系统的用例图，E-R 图进行了绘制，有了这些之后，我们就可以进行下一步操作。接下来我们将对于系统实现展开论述

第4章 考勤签到系统功能实现

4.1 系统实现

4.1.1 系统前端实现

在通过 `vue-cli` 脚手架创建的项目里，我们进入到 `src` 目录，分别按照设计的模块创建好我们会用到的组件。因为采用前后端分离编写，数据处理可以在前端也可以在后端实现。下面我将重要的功能与数据出炉每个组件对应着前端页面的每一个功能，因为我们的功能有登录模块、数据可视化模块、签到模块、学生管理模块、教师管理模块，分别对应 `login`、`attendance`、`student`、`teacher` 这些页面，其余还有一些页面实现了其他不重要的功能将会在下面陈列：

1) `login.vue` 实现登录模块

这个页面主要是用来处理用户登录的，页面上使用了全浏览器视图大小的自适应视屏背景，一个登录框，上面能够选择教师或者学生登陆，然后还有读取 `WebStorage` 的功能。下面将列举处理学生登录的实现例子，教师的登录也和学生登录逻辑差不多，其中一小段学生登录的关键代码如下：

```
if (identity === '学生') {  
    let qstarttime = GetQueryString('time')  
    //通过 axios 发送 POST 请求，data 是从页面里面输入的用户手机号和密码  
    this.$axios.post('/api/user/studentLogin', data).then((response) => {  
        if (response.data) {  
            console.log(response.data[0])  
            //webStorage 存储学生信息和登陆状态  
            window.localStorage.setItem('studentInfo',JSON.stringify(response.data[0]))  
            window.localStorage.setItem('isLogin','1')//本地存储设置登录状态为登录  
            localStorage.setItem('userName',response.data[0].sname)//存储登录者的姓名
```

`this.$store.dispatch('changeUserName',response.data[0].sname)//更`
 改 Vuex 里面的数据

```

    this.$store.dispatch('changeLogin',true)
    let did =
JSON.parse(window.localStorage.getItem('studentInfo')).sno
    setTimeout(function() {
        this.$router.push({//当我们发现浏览器本地存有数据，就将页
面 push 到学生管理页面，并附带参数
            path: `/student?did=${did}&time=${qstarttime}
        })
        }.bind(this), 500)
    } else {
        this.err()
    }
    })
  }
  // console.log('submit!')
} else {
  // console.log('error submit!!')
  return false
}

```

这里，我们通过页面传参，传递到学生页面，然后学生页面通过\$route 对象去获取参数，如果参数存在，则登录成功，如果参数不存在，那么就重新返回登录页面。

2) attendanceList.vue 实现数据可视化

这个页面主要展示的是一个图表。里面获取到后台签到和迟到人数的比，然后在该页面渲染一个圆饼图。主要用于展示一周的签到/迟到的情况，实现了数据可视化，方便统计

3) index.vue 实现总览签到情况

这个页面主要展示一周内每天签到人数的柱状图，从周一到周五，数据是通过 axios 异步从后台获取，然后将这些数据通过 Vue 的双向数据绑定功能实时传递

到 Echarts 对象里面，动态渲染柱状图，能够直观的感受得到一周内签到人数的变化比值。其中具体的数据获取和渲染部分的代码如下所示：

//按日期，签到人数

```
drawPie: function () {
    this.myChart = echarts.init(document.getElementById('pie'));
    let data={
        day1:'%04-08%', day2:'%04-09%',
        day3:'%04-10%', day4:'%04-11%',
        day5:'%04-12%',
    }
    var tempArr = []
    this.$axios.post('/api/echart/echartTimes',data)
    .then((res) => {
        for(let i=0; i< res.data.length ; i++){
            tempArr.push(res.data[i].times)
        }
        let str = JSON.stringify(tempArr)
        tempArr = JSON.parse(str)
        // 绘制图表 异步编程的重要性，小心数据获取不到
        var option = {
            title: {
                text: '过去一周签到情况'
            },
            tooltip: {},
            xAxis: {
                data: ['星期一', '星期二', '星期三', '星期四', '星期五']
            },
            yAxis: {},
            series: [{
                name: '签到人数',
                type: 'bar',
                data: tempArr
            }]
        }
    })
}
```

```

    }}
  }
  console.log(option.series[0].data)
  this.myChart.setOption(option);
}).catch((err) => {
  console.log(err)
});

```

以上页面实现了数据的可视化，主页能够展示一周内签到的比例。

4) notFound.vue

这个是处理系统内未定义的页面，也就是 404notFound 页面。

5) qingjia.vue 实现请假功能

这个页面里有一个响应式的 form 表单，学生需要在里面填写请假时间，请假原因，请假类型等，然后可以提交到后台。该页面使用了 HUI 的 \$message 组件，绑定在提交按钮上，当我们验证表单并通过 axios 传递到后台并接受到后台的反馈后，页面会对学生的这次请假行为进行一个小提示，这会让用户体验更加良好。

6) student.vue 为学生展示页面

这个页面做成响应式，方便手机端和电脑端能够看到风格一样的学生自己的签到界面，这上面集成了几个按钮，学生可以点击这些按钮进行不同操作，中间还有一个轮播图。

7) teacher.vue 为教师管理页面

这个页面是教师的管理页面，老师可以设置签到时间，然后在右侧的 tab 页面里面，还能查看学生的签到情况、请假情况，还可以发布可缩放的二维码，具体二维码功能的实现已经在上一节进行了说明，这里将不再赘述。

8) userCenter.vue 用于展示学生签到信息

这个页面主要是学生使用，里面详细展示了该学生的所有签到情况。

4.1.2 系统后台实现

后台主要是使用 express 框架，启动一个服务器，然后分模块编写后台 API。首先，建立一个 app.js 的文件夹，然后在其内部实例化一个 express 对象，这个 express 对象需要经历一些第三方中间件和自己编写的中间件，然后再暴露 API 于

启动的服务器上，然后再让这个 `express` 对象监听一个端口，实现前后端两个服务器的对接。

开发过程中能够使用 `node` 本身能够搭建的服务器，不过在项目上线后需要使用 `Nginx` 反向代理这个服务器，增强其性能。要实现基于 `express` 的服务器代码如下：

```
const express = require('express');
const app = express();
app.listen(3000)
```

以上就搭建好了一个 `localhost:3000` 的端口。

现在我们应该对数据库设计和生成，我们需要建立如下的数据库表：

分别是 `dictionary` 班级信息表，`qdrecord` 签到信息记录表，`qingjiarecord` 请假信息记录表，`starttime` 系统初始化签到时间表，`student` 学生信息表，`teacher` 教师信息表，`teacherclass` 教师管理班级表，这些表的设计，实现方式我们已经在上一个章节：系统功能详细设计中已经描述了。这里我们只需要在数据库中使用 `SQL` 语言实现这些表格就完成了

然后进入后台文件建立 `SQLmap.js` 文件，编写拿取数据的 `SQL` 命令，这里类似一个数据库操作的库，我们将所有需要用到数据库的地方进行封装，其中一小部分命令展示如下：

```
student: {
  query: 'select * from student where sphone = ? and spassword = ?',
  record_qd: 'INSERT INTO
qdrecord(studentno,qdate,qtime,qstarttime,qteachermsg) VALUES (?,?,,?,?)',
  querybyname: 'select s.sname,s.ssex,s.sphone, q.qdate,q.qteachermsg'
  +' from qdrecord q,student s where q.studentno = s.sno and s.sno= ?',
  // 'select * from qdrecord where studentno = ?'
  qingjia: 'INSERT INTO qingjiarecord(studentno,sname,qtype,qres,qdate,teacherno)
VALUES (?,?,,?,?,34)',
},
```

然后针对前端传过来的数据，我们的处理方式是暴露的 `api` 地址确定使用哪一个 `SQL` 命令，以及如何处理数据，例如签到这个请求，前端会传他的信息和签到时间：

```

router.post('/recordQd', (req, res) => {
    var sql = $sql.student.record_qd;
    var params = req.body;//传入四个数据,
    console.log(params);
    conn.query(sql,
[params.studentno,params.qdate,params.qtime,params.qstarttime,params.qteachermsg],
function(err, result) {
    if (err) {
        console.log(err);
    }
    if (result) {
        jsonWrite(res, result);
    }
    })
});
module.exports = router;

```

然后通过 `express` 对象 `use` 这个中间件，并挂在在端口上，那么就实现了一个 API:

```
app.use('/api/record', recordQd);
```

到这里后端其他功能不需要过多赘述。

4.1.3 系统二维码生成以及移动端扫码签到代码实现

当教师在教师管理页面进行设置之后，会将参数传递到二维码的接口当中，当我们确认教师在设置、或者使用系统默认设置签到时间，且在选择了签到班级的情况下，我们会将班级 id、学生入口、最迟签到时间获取到，并且传递到我们的二维码生成的接口函数当中去，代码如下：

```

if(this.startTime){
    this.$options.methods.createQr(cid,this.qrUrl,this.startTime)// methods 中调用另
外一个声明函数要使用 this.$options.methods.函数名()
}

```

这样，我们页面中与这个接口关联的 SVG 标签就会生成一个二维码，然后学

生拿出手机打开浏览器，使用浏览器的扫码功能，扫描二维码，他就会接收到一个参数，这个参数含有：班级 Id、入口地址、最迟签到时间。

页面会跳转到入口地址，判断浏览器本地存储里面有没有用户信息，有的话就免去登录，直接跳到签到环节，这里的代码如下：

```
if(!window.localStorage.getItem('teacherInfo')){//判断是否有信息
    this.$router.push('/')//没有就到首页
}else{//有信息就获取
    this.teacherName=JSON.parse(window.localStorage.getItem('teacherInfo')).teacherName
}
```

学生进入到管理页面之后点击签到，系统会生成一个当前时间对象，然后将其与传递过来的时间参数进行对比，如果比最迟时间早，那么就是正常签到；如果迟于最迟签到时间，那就是迟到。代码如下：

```
if(nowDate.getHours() < qstartHour){
    data.qteachermsg='正常'
}
else if(nowDate.getHours() >= qstartHour)
{
    if(nowDate.getMinutes() > qstartMinutes){
        let lateTime = nowDate.getMinutes() - qstartMinutes
        data.qteachermsg = '迟到'+lateTime+'分钟'
    }
    else{
        data.qteachermsg = '掐点到'
    }
}
```

针对这段代码，我们画一个流程图，方便理解，如下图 4.1 签到时间判定代码流程图所示：

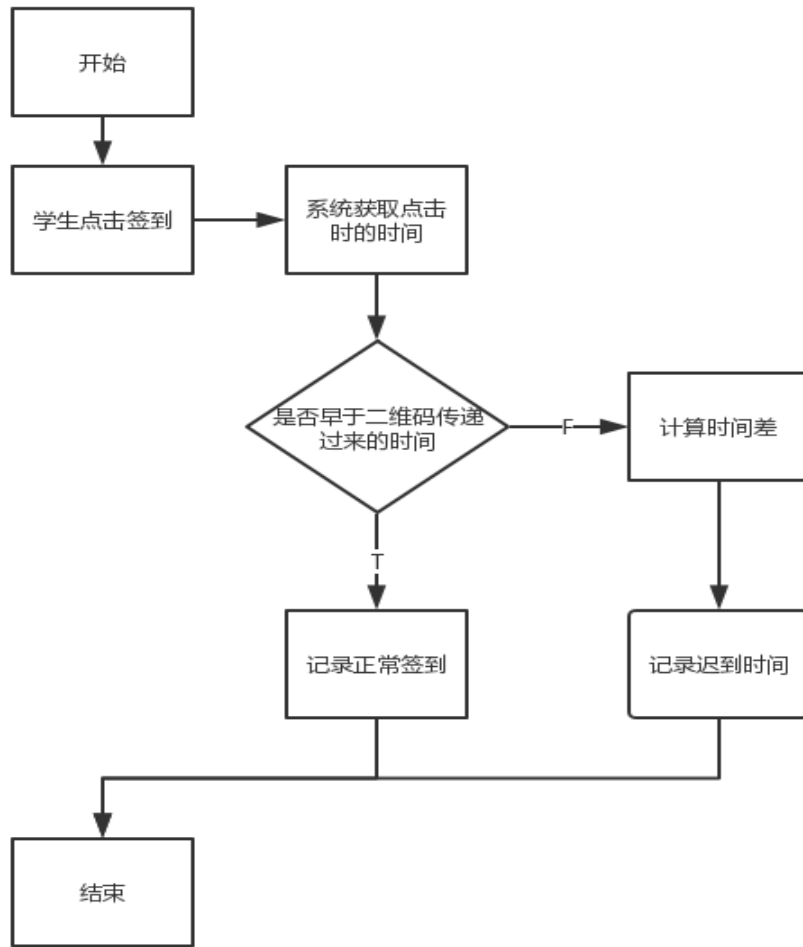


图 4.1 签到时间判定代码流程图

现在,我们只需要把这些数据传递给后台就行了,通过后台写入数据库即可,这个功能已经在上一个点叙述过了,这里不再赘述。

4.1.4 跨域

由于浏览器的同源策略,前台问到后台数据的。当我们使用 axios 对后台 ip 请求数据的时候,浏览器会报错,这就是同源安全策略导致的问题。所以我们需要在后台全局发送的数据里设置允许跨域的白名单,用中间件来表示,这里我们设置的是所有皆为白名单:

```
app.all('*', function(req, res, next) {
```



```
res.header("Access-Control-Allow-Origin", "*");//项目上线后改成页面的地址
res.header("Access-Control-Allow-Headers", "X-Requested-With,Content-Type");
res.header("Access-Control-Allow-Methods","PUT,POST,GET,DELETE,OPTIONS"
);
next();
});
```

4.2 本章小结

本章节主要讲述了对基于二维码的考勤签到系统的设计与实现，分前后端的开发步骤进行了详细的描述，前端分页面描述了页面上的逻辑，并讲述了如何与后台进行交互以及数据渲染，还讲述了 QRcode.js 库，讲述了二维码 API 的使用，二维码传参，后端则讲述了数据库表的设计与实现，以及如何获取传输字段，以及如何跨域，最后再对系统关键功能的实现进行了讲述。

第 5 章 系统测试

5.1 测试以及结果

5.1.1 登录模块测试

登录模块测试用例如下表 5.1 所示：

表 5.1 登录测试用例

编号	前提条件	用户电话	用户密码	预期结果	实际结果
001	选中学生入口	18983845705	123456	成功登入学生页面	成功登入学生页面
002	选中学生入口	22222222222	123	成功登入学生页面	成功登入学生页面
003	选中学生入口	33333333333	空	登录失败	登录失败
004	选中学生入口	空	123	登录失败	登录失败
005	选中学生入口	空	空	不能为空	登陆失败
006	选中教师入口	18888888888	1	成功登入到教师页面	成功登入到教师页面
007	选中教师入口	18888888888	空	登录失败	登陆失败
008	选中教师入口	空	1	登录失败	登陆失败
009	选中教师入口	99999999999	123	学生账号登录失败	应以教师账号登录
010	登录	空	空	点击退出，用户信息删除返回登录页	成功返回到登录页面。
011	教师发布二维码，学生第一次扫码	22222222222	123	页面进入登录界面输入账号密码才能登入界面	移动端扫描输入账号密码进入学生管理页面

012	教师发布 二维码， 学生第二 次扫码	22222222222	123	页面免登录 直接进入学 生管理页面	移动端扫码 直接进入管 理页
-----	-----------------------------	-------------	-----	-------------------------	----------------------

5.1.2 扫码签到模块测试

测试编号：2.1 生成二维码

测试用例：教师进入教师管理页面设置时间、班级，然后发布二维码。

测试方法：目测。

测试结果：页面二维码生成成功。

测试编号：2.2 扫码输入登录

测试用例：学生扫码，浏览器输入账号密码，点击登陆。

测试方法：目测。

测试结果：登陆成功。

测试编号：2.3 扫码免输入登录

测试用例：与 2.2 中使用同一手机、同一浏览器，扫码。

测试方法：目测。

测试结果：无需手动登陆，直接跳转签到界面。

测试编号：2.4 签到

测试用例：任意一个学生账号进入到学生管理页面，然后点击签到按钮。

测试方法：目测。

测试结果：页面弹框，提示成功签到，然后点击个人签到信息，发现多了一条数据。

5.1.3 请假模块测试

测试用例：任意一个学生账号进入学生管理页面，然后点击请假按钮，进入

到请假页面，然后输入数据，提交

测试方法：目测。

测试结果：弹框，提示成功提交，然后返回教师管理页面查看学生请假情况，可以发现多了一条数据，请假成功。

5.1.4 性能测试

使用 JMeter 这款开源的小巧的性能测试工具，在 JMeter 新项目中添加一个“Threads(Users)”线程组，这里配置虚拟用户数量。我们配置了 10 个线程，也就是 10 个虚拟用户，然后设置准备时长为 1，也就是每秒钟启动 10 个线程。然后我们设置了循环次数 100，那么每个线程发送 100 次请求，总请求数为 1000。然后录入登录操作的脚本，定义好用户电话和密码参数，然后启动脚本，让其虚拟用户重复执行登陆操作。

当我们等待了数分钟之后，进入 JMeter 的结果界面进行查看。

10 个虚拟用户，总共 1000 次访问，平均 1600ms，最小响应时间是 1320ms，最大响应时间是 2100ms，错误率为 0。所以得到性能测试结果为合格。

5.1.5 测试结果

测试结果：经过测试系统的主要功能模块的功能能够实现，功能合格，数据展示正常，各个模块的测试结果基本满足测试期望。

然后在性能测试中用户登陆操作的平均响应时间远低于 1s，所以性能测试也是合格。

所以最终结果是测试合格。

5.2 项目部署与界面 UI

5.2.1 项目部署

前端使用 `npm run build` 对代码进行打包之后，然后将前端文件放入 Nginx 的

www 下，并对 nginx.conf 进行配置，然后就可以通过配置好的 ip 和端口进入到前端项目中。后端项目使用 Node.js 启动内部集成的 http 服务器，将接口暴露出来，再将 MySQL 数据库启动起来，这样前后端两个项目就部署成功了，且能够通过 Nginx 设置的地址对项目进行访问。

5.2.2 系统界面展示

1) 登录页面

该页面为教师端和学生端的 pc 入口，学生可以选择学生登录入口，不过通过 pc 端进入无法签到，教师可以选择教师登录入口，登录成功后会进入到教师管理页面。

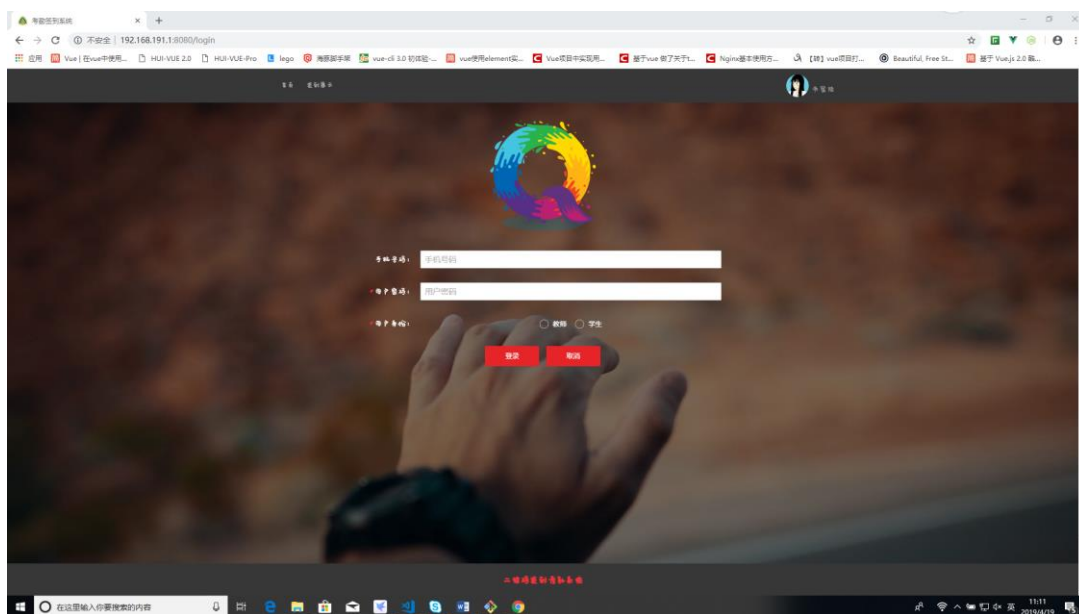


图 5.1 系统 PC 端登录页面

2) 教师管理页面

该页面是教师登录成功后，教师进入系统设置页面，教师可以在此页面设置签到的一部分参数，例如时间和签到地址等。



图 5.2 PC 端教师管理页面

3) 教师管理页 2

该页面是教师查看学生请假记录，前提条件是教师登入管理系统。



图 5.3 PC 端教师管理页面 2

4) 教师管理页 3

该页面是当教师登入，进入教师管理系统后，教师可以查看学生出勤记录，里面包含了签到时间和迟到时间，以及签到名单

扫码签到

出勤记录

请假记录

Q 查询所有签到信息

姓名	性别	电话	签到时间	签到状态
██████	1	██████████	2019-04-10T07:36:58.000Z	正常
██████	1	2222222222	2019-04-10T07:45:44.000Z	正常
██████	1	18983845705	2019-04-10T07:45:44.000Z	迟到54分钟
██████	1	2222222222	2019-04-11T07:45:44.000Z	正常
██████	1	3333333333	2019-04-11T07:45:44.000Z	正常
██████	1	5555555555	2019-04-11T07:45:44.000Z	正常
██████	1	4444444444	2019-04-11T07:45:44.000Z	正常
██████	1	5555555555	2019-04-11T07:45:44.000Z	正常

图 5.4 PC 端教师管理页面 3

5) 教师管理页 4

该页面是教师选择班级，当教师登入且设置参数后触发二维码生成，生成二维码的依据是系统设置和班级 Id。



图 5.5 PC 端教师管理页面 4

6) 数据可视化页 1

该页面是首页，用于展示一周每天签到人数，并用图表，将数据进行可视化

处理。

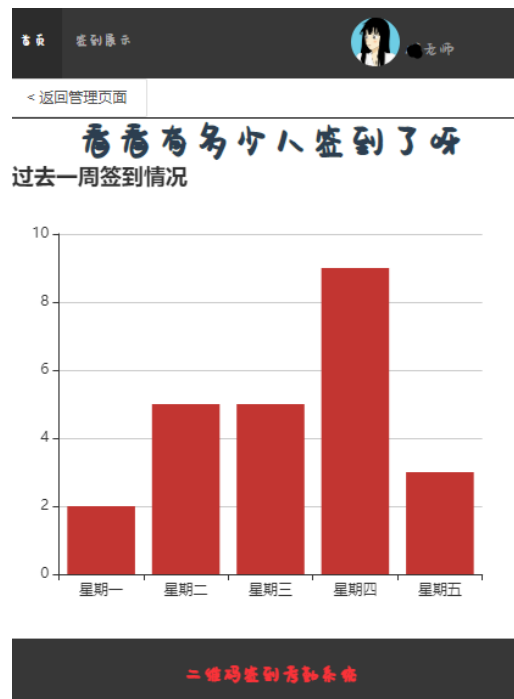


图 5.6 数据可视化页 1

7) 数据可视化页 2

该页面展示了数据库中所有数据的正常签到和迟到比。

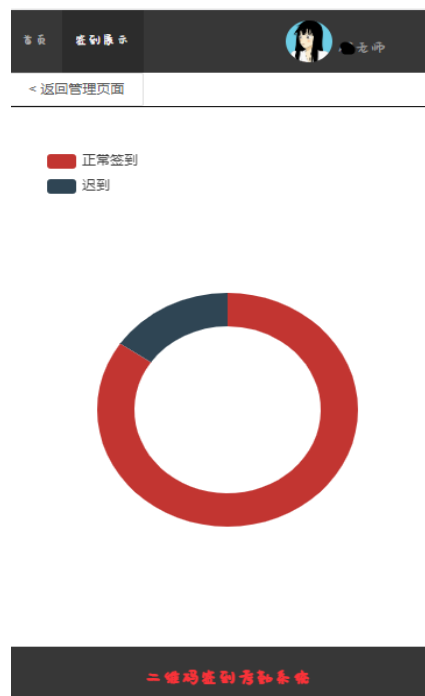


图 5.7 数据可视化页 2

8) 手机端学生页面

该些页面分别对应学生第一次使用该系统，扫码签到的时候，需要输入手机号和密码进行登录，然后是学生的请假表单页面和学生管理页面，学生管理页面里面有签到按钮，学生在接受到参数后，系统在断定有参的情况下才可以正常签到。

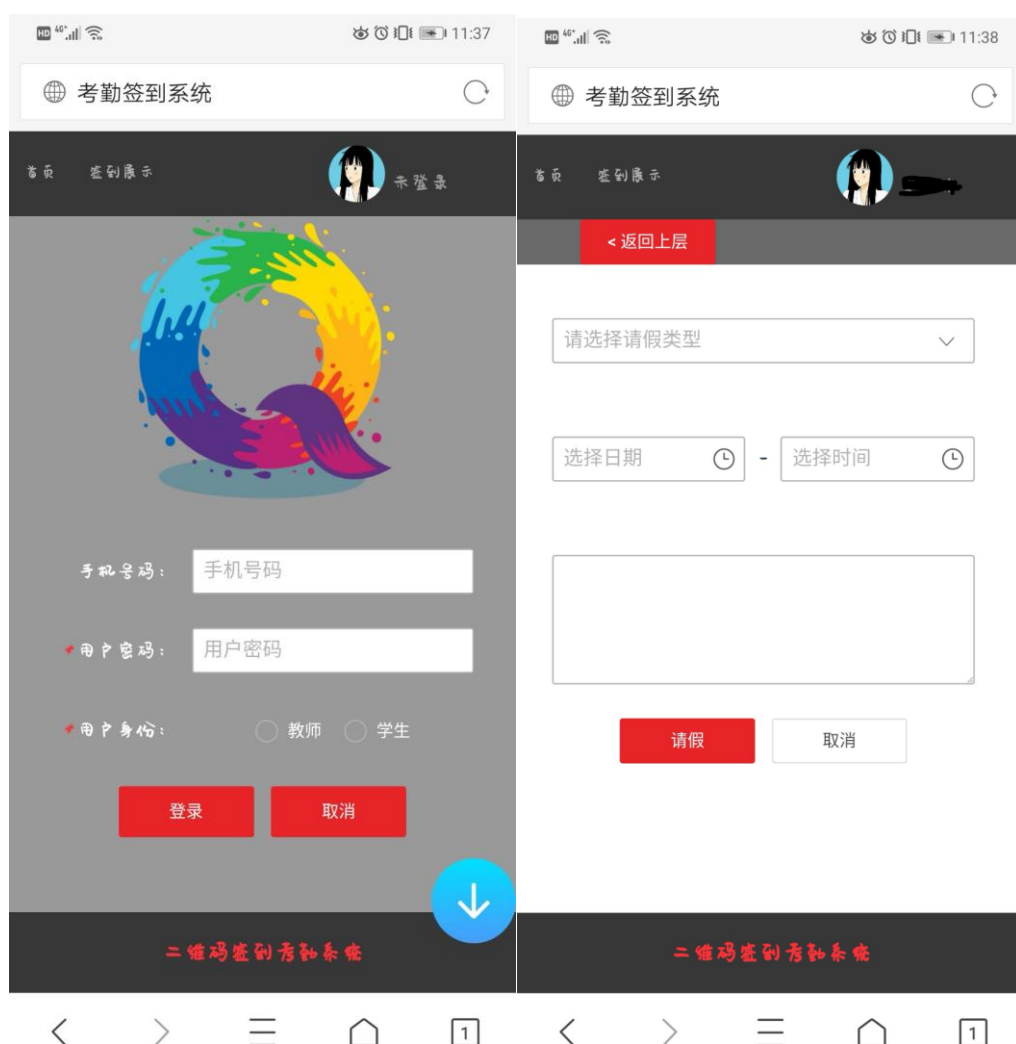




图 5.8 学生端的登录、请假、管理页面

5.3 本章小结

本章节主要是对系统进行了功能测试和性能测试，最后得到结果是测试结果满足期望值，整个项目基本功能已经完成了，然后性能也有保障。最后我们对项目进行了部署，项目前端使用 Webpack 集成的服务器将代码进行启动，也可以使用 Nginx 对项目进行部署，然后后端使用 node.js 的 http 服务器对代码后台进行了部署。最后将实现效果通过项目截图列举在本章节中。

第 6 章 总结与展望

6.1 总结

项目和论文经历了几个月，其中还有在外实习的原因，且公司是 996 文化，边工作边写论文非常苦难，这段时间真的造成了我前所未有的窘迫情形，也是时间未规划好而造成的。一切都非常匆忙，幸亏有导师给我提供帮助和思路，能让我的毕业设计和毕业论文能够顺利进展下去。本文初版写完断断续续大概花费了一个半月的时间，下面将对整篇文章进行总结。

首先在第一章节，对于为什么去设计与实现这样一个系统展开了论述。就是为了绿色环保，方便快捷的解决传统手工纸质签到带来的各种不便，所以展开了对于本系统的设计与实现。

第二章节，开始对本系统将要用到的 QRcode 和全栈技术进行了分析，介绍了二维码的来历和 QR 码的原理，然后介绍了以前端框架 Vue 为主的一些前端技术、以 express 框架为主的后端技术，为项目的开发打下技术基础。

第三章节，主要对系统进行了设计，显示模块，二维码签到流程等进行了总体设计，然后对前端页面实现过程进行了描述，对后台数据操纵进行了描述，对一些典型的功能进行了描述，最后对界面实现进行了展示。

就这样，一个拥有签到考勤请假基础功能的系统就完成了，可是由于比较忙还有时间局促，以及个人能力有限的原因，代码有很多地方写的很乱，还有 BUG。但是已经勉强达成了目标

6.2 展望

没有几个项目经验，没有太多工作经历，我的技术能力没达到期望值，但是幸好有导师全程帮助我们解决技术上的问题，为项目发展提供思路。我通过此次毕业设计了解到了学习的重要性。只有学习到位了，工作才能顺利的进展下去，

对于我的这个毕业设计项目，希望未来能够努力将我的技术能力提升，然后能够将本项目实现，免费开源给大学课堂。

这几个月让我学到了不少知识，从收集资料到开会探讨问题。从开展项目设计工作到实施编码。我不仅仅提高了专业技能，还学会了做事情要脚踏实地，踏踏实实的做事，认真勤劳，勤动脑才能另辟蹊径。

相信将来无论是生活，还是工作当中，只要回忆起这段难忘的会议都会像给自己打了一针强心剂。世界上没有过不去的坎儿，只要勇于探索，忘记烦恼，认真对待，努力克服困难，一切将会像浮云一样慢慢散去。随着科学技术的进步，我们也需要与时俱进，不断去学习新技能，才能不被这个社会所淘汰。只有技术过硬，才会顺风顺水，遇到问题才会迎刃而解。就像做项目一样，我们必须学会项目上的实战优化后才能逐步沉淀到下一个新版本，耐得住性子，一个一个应用的做，将应用做好后，再结合架构+组件，做了多个应用后才能实现一个平台软件，而你的每一个小软件都为这个平台软件添砖加瓦，做了自己的贡献！

结论

本论文所设计的系统能够以绿色环保的方式去实现快速便捷的签到，比传统的纸质签到方式要高效不少。因为二维码原生编码实现十分艰难，再加上本人能力有限，所以在二维码生成方式上，我采用了一个二维码生成工具：`qrcode.js`。

在项目设计过程中，开始很难去设想到怎样去通过扫码去签到。后来因为想到了浏览器的本地存储这个功能，就打算利用本地存储直接存储用户信息，从而让学生扫码的时候直接从手机浏览器的本地存储里面获取到用户的数据，然后加上二维码传递过来的参数，直接定向到签到页面，这个功能就可以勉强实现了。

整个系统是前后端分离开发，结构较为清晰明了。开始走了些弯路，不过后来，边实习边做毕设，然后就慢慢做了出了，能够勉强实现开题报告中定下的目标，能够实现便捷的签到，提供考勤统计等功能。UI 框架使用了实习公司的 UI 框架，不方便拷贝出来。项目多多少少还是有很多缺陷，加上我并没有服务器，电脑 IP 每天都在变化，所以这个项目在演示前还需要很多操作，后续会想办法解决。

致谢

毕业近在眼前，有很多感叹，毕业设计基于二维码的考勤签到系统的设计与实现是代老师为我出的题目，对于二维码部分，当时代老师要我把二维码实现出来，我还觉得蛮有信心。后来经过查看文献和现成的二维码库，发现想要自己来写个二维码不是想象中的那么简单，所以后期着重把题目重点从二维码转移到应用上面来，因为我是搞 Web 前端这块的，对应用还是比较拿手。非常感谢代老师在这个期间对我的指导。

青春如此短暂，我在写论文期间，每天都是白天在外上班，晚上回到出租屋内写论文，我既希望论文写完，又不希望论文写完，也许这就是人生吧，在纠结当中，白驹过隙，最终还是得面对毕业那一刻。996 的工作日下，经常写论文到凌晨，所以质量粗糙，请多多包涵。

感谢大学期间的辅导员-杨莉，让我们大学生活有序。

感谢指导老师-代老师，让我的毕业论文顺利完成。

感谢大学期间教过我的老师，让我明白奋斗的意义。

感谢大学期间陪伴过我的同学，愿下半生还能和你们保持联系。

参考文献

- [1] 王鹤琴,朱珍元.基于 MVVM 模式的 Web 开发研究[J].菏泽学院学报,2019(02):7-13.
- [2] 魏钢.Web 前后端分离模式下 Spring MVC 在高职 Java Web 教学中的研究[J].福建电脑,2019,35(02):107-108.
- [3] 牛仁腾. 基于 Vue.js 的表单可视化构建系统的设计与实现[D].华中科技大学,2019.
- [4] 孙一笑,张玉军,孙宇成,吕卉元.基于 WebAPI 前后端完全分离的软件开发模式[J].信息与电脑(理论版),2019(06):96-97.
- [5] 贾丛丛. 基于 MVVM 的 Web 前端响应式框架的研究与集成[D]. 云南大学硕士学位论文, 2016.
- [6] 刘红卫.利用 Node.js 开发前后端分离的系统——以图书馆地方文献系统为例[J].天津科技,2018,45(07):67-70.
- [7] Vue.js 实战[M]. 清华大学出版社 , 梁灏, 2017.
- [8] 徐頔,朱广华,贾瑶.基于 VueJs 的 WEB 前端开发研究[J].科技风,2017(14):69.
- [9] 许会元, 何利力.NodeJS 的异步非阻塞 I/O 研究[J]. 工业控制计算机第 3 期, 2015.127-129.
- [10] 许正义. 基于 NodeJs 的成果管理辅助系统设计与实现[D]. 江西师范大学, 2018.
- [11] 张晓颖.试析基于 Node.js 的前后端分离框架的实现[J].计算机产品与流通,2018(10):24.
- [12] 杜艳美,黄晓芳.面向企业级 web 应用的前后端分离开发模式及实践[J].西南科技大学学报,2018,33(02):83-87.
- [13] 王道大. SVG 与 Ajax 在工业远程在线实时监控系统中的应用研究[D]. 昆明理工大学, 2016.
- [14] 麦冬, 陈涛, 梁宗湾. 轻量级响应式框架 Vue.js 应用分析[J]. 信息与电脑(理论版), 2017. 58-59.
- [15] 邓雯婷. 基于 Vue.js 构建单页面 GIS 应用的方法研究[J]. 科技创新与应用第 14 期,2018 .5-7、10.

- [16] 朱二华. 基于 Vue.js 的 Web 前端应用研究[J]. 科技与创新第 20 期. 2017, 119-121.
- [17] 黄有福. HTML5 WebStorage API 与.NET 会话机制探索[J]. 电脑知识与技术 (学术交流)第 3X 期, 2013, 2102-2104.
- [18] Paul Krill,Paul Krill. Vue.js lead: Our JavaScript framework is faster than React[J]. InfoWorld.com,2016.
- [19] Wei Yu,Yong Xu. Research on the application of HTML5 in the development of mobile internet[J]. Journal of Intelligent & Fuzzy Systems,2018,34(2).
- [20] 张灵凤, 冯锋, 黄恒. 二维码技术的生成原理的分析与研究[J]. 超星期刊第 17 期第 25 卷.
- [21] Martínez Lascorz, Jorge. Seguridad web con NodeJS en un sistema de gestión de horarios laborales[D]. Universidad Politécnica de Madrid. 2017.
- [22] Brophy; Matthew Benjamin. OWL-PL: A presentation language for displaying semantic data on the web.[D]. Lehigh University.2010
- [23] 罗飞. 基于二维码 QR 码技术的手机签到系统的设计与实现[D]. 华中科技大学硕士学位论文. 2013
- [24] 阮一峰. ECMAScript 6 Primer ES6 标准入门(第 3 版)[S]. 2017.9
- [25] Shuguang Yuan,H.C. Stephen Chan,Zhenquan Hu. Implementing WebGL and HTML5 in Macromolecular Visualization and Modern Computer-Aided Drug Design[J]. Trends in Biotechnology,2017,35(6).

附录 英文翻译

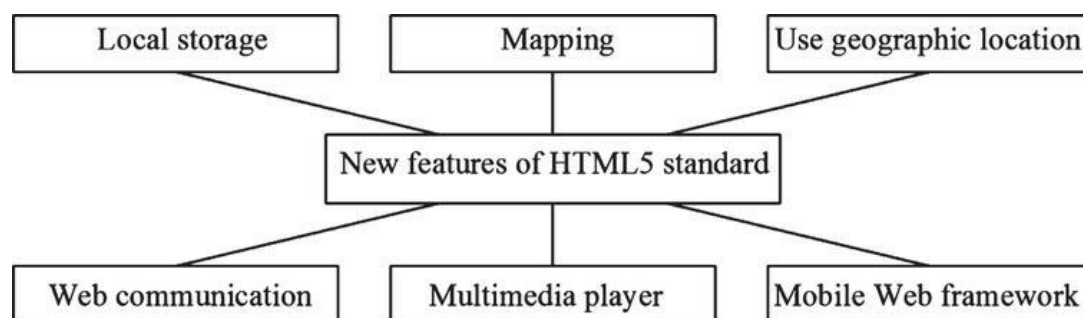
一、英文原文

Research on the application of HTML5 in the development of mobile internet

Yu, Wei; Xu, Yong

Application of HTML5 in mobile internet

HTML5 is a new generation of HTML standard introduced by W3C [9]. Prior to the advent of W3C, the HTML standard was the HTML4.01 standard in 1999 and the XHTML1.1 standard in 2001 introduced by W3C. With the rapid development of Internet technology and the rapid rise of the mobile Internet, HTML4.01 standards and XHTML1.1 standards that are not upgraded for many years are unable to adapt to the needs of the development of the Internet in the new situation. Therefore, the upgrading of the HTML standard is getting louder and louder [10]. So in 2008, W3C released the first draft of the HTML5 standard, which was used to replace the outdated HTML4.01 standards and XHTML1.1 standards. As a new Internet HTML standard, HTML5 standard is a set of standards including HTML5, CSS3 and JavaScript [11]. Its most important feature is that it can reduce the demand for plug-ins in the rich network application services when viewing the browsers, and it can increase the standard set for enhancing network applications [12]. Specifically, the HTML5 standard adds not only the new syntax features such as <video>, <audio>, and <canvas>, but also the contents of the scalable vector graphics, which are mainly used to add and process the multimedia and image content in the browser. In addition to the new content, HTML5 also makes new attributes and elements of the redefinition, standardizations, deletions and modifications, etc. based on the previous generation of HTML standards, so that it is more in line with the needs of the new era of network development. In October 29, 2014, W3C announced the completion of the HTML5 standard specification.

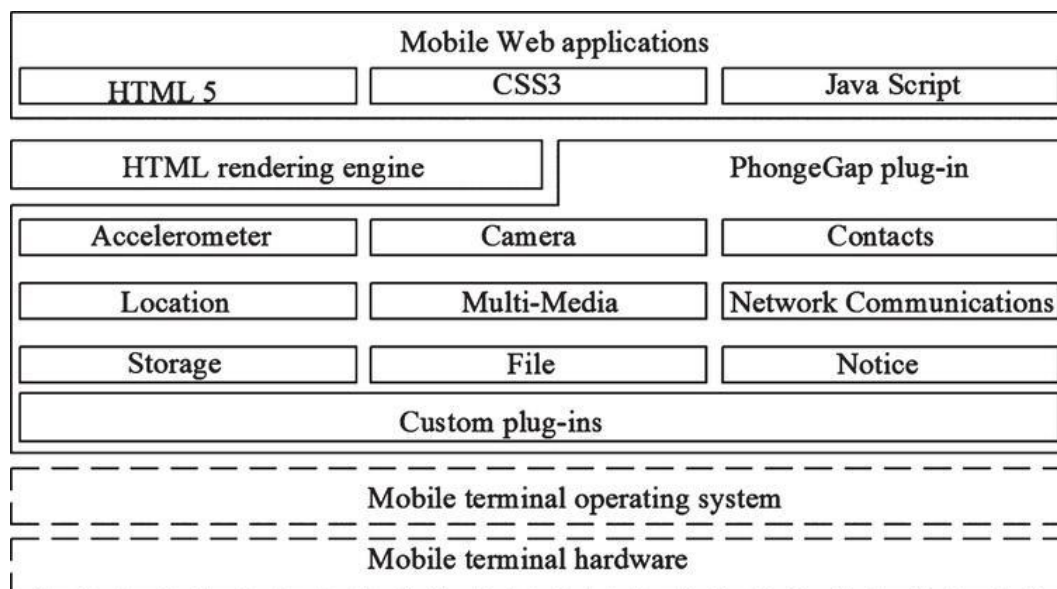


After the completion of the HTML5 standard, it has been widely concerned by the developers of mobile Internet. In the HTML5 standard, in order to deal with the problems in the application of Web in the mobile Internet, HTML5 standard provides many new technical features [13]. These technical features include local storages, graphics, multimedia playbacks, Web communications, and uses of geographical locations and mobile Web frameworks. These new features enable mobile Web applications to achieve and enrich a variety of network applications, such as graphics, mobile devices, offline uses, as well as browser video playbacks on the PC platform, so as to achieve the functions and applications [14]. In addition, the HTML5 standard overcomes the problem that the browser relies too much on the plug-in, so that the mobile Web devices can provide more conveniences and conditions in the limited memory spaces [15]. In addition, in the mainstream mobile operating systems, such as the mobile phone, the Webkit kernels used by the browsers can support the HTML5 standard and CSS3 standard, so that the same application software can be used across platforms. It is rich in applications, and because the new characteristics and functions of HTML5 save storage resources, cross platforms and obtain easy version controls, Web mobile Internet applications will become the mainstream in the future application.

A variety of Web applications in the mobile Internet depend heavily on the operating system of the hardware. In the current market, there are three main known operating systems, namely, Android, IOS and Windows. The hardware operating systems are different, so that the mobile Web application developers will make development on the same application software, so as to ensure the use of software crowds, and the costs of development and maintenance of application software are rising rapidly [16]. With the release and large-scale application of HTML5 standard, the mobile Web application software which can be used in cross platform has been widely used in the mobile Internet, such as the mobile version of Baidu maps, Iqiyi and Youku.

They all use the new features of HTML5 to achieve audio and video playbacks [17]. These Web applications based on the development of HTML5 standards have a good performance in the use of performance and user experience. The major companies have gradually abandoned the Flash support for mobile devices, and have studied a variety of cross-platform applications based on HTML5 standards, so that application of HTML5 in the mobile Internet is wider [18]. However, based on the HTML5 standard, in the use of hardware features of mobile device, mobile Web application software needs to be achieved through Device API of the operating system browser in mobile devices. Therefore, the browser needs to have good HTML5 support capabilities. However, the browser of current mainstream operating systems has different opinions for HTML5 standards, so that HTML5 can't be used in some browsers.

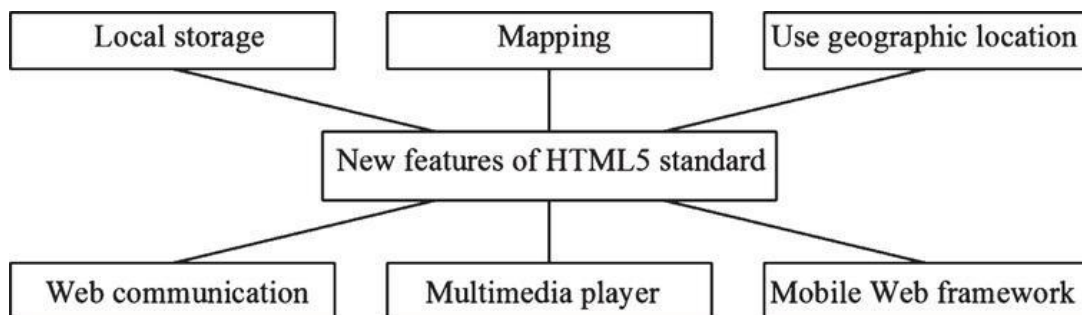
At present, the mainstream application of HTML5 in the mobile Internet is the mixed application (Hybrid APP), which is to overcome the problem of some browsers that can't support the HTML5 standard [19]. The so-called hybrid application refers to the mobile Web application and operating system that can take a compromise between their own applications. By combining the new features of HTML5 standards and the needs of mobile Web applications, the hybrid APP (Hybrid) has the advantages of mobile Web applications and operating system applications [20]. Through the application of mixed methods, software developers use HTML5 technology to overcome the system's own software platforms and fragmentation issues. The browser that can't fully support the problem of Device API can be solved through the mixed application plug-in of the JavaScript technology in HTML5 standards, so that the various characteristics of the HTML5 standard can be fully played. In the current market, there are many kinds of development frameworks of mixed applications, and the current mainstream development frameworks are PhoneGap, AppMobi, AppCan and Titanium. Apart from mixed applications, HTML5 applications in the mobile Internet include mobile Web App and Web application store, in which the mobile Web App is not as good as the hybrid application in the compatibilities of HTML5, cross-platforms and development costs



二、英文翻译

HTML5 在移动端的应用

HTML5 是 W3C 引入的新一代 HTML 标准。在该标准出现之前，HTML 标准泛指 1999 年的 HTML4.01 标准和 W3C 引入的 2001 年的 XHTML1.1 标准，随着互联网技术的迅速发展和移动互联网的迅速崛起，HTML4.01 标准和 XHTML1.1 标准经过多年的不升级已经不能适应发展的需要。因此，HTML 标准的升级变得越来越迫在眉睫。因此，在 2008 年，W3C 发布了 HTML5 标准的初稿，用于取代过时的 HTML4.01、XHTML 1.1 标准。作为一种新的互联网 HTML 标准，HTML5 囊括了很多东西，包括 HTML5、CSS3 和 JavaScript。它最重要的特征是，在浏览网页时可以减少对富网络应用程序服务插件的需求，并且可以增强网络应用程序的标准集。具体来说，HTML5 标准不仅添加了新的语法功能，如<video>、<audio>和<canvas>，还添加了可缩放矢量图形的内容，这些内容主要用于在浏览器中添加和处理多媒体、图像内容。除了新的内容外，HTML5 也在上一代 HTML 标准的基础上做出了许多新的定义、新的标准化以及删除和修改等新的属性和元素，使其更符合网络发展新时代的需要。2014 年 10 月 29 日，W3C 宣布完成 HTML5 标准规范。以下是 HTML5 带来的新内容：

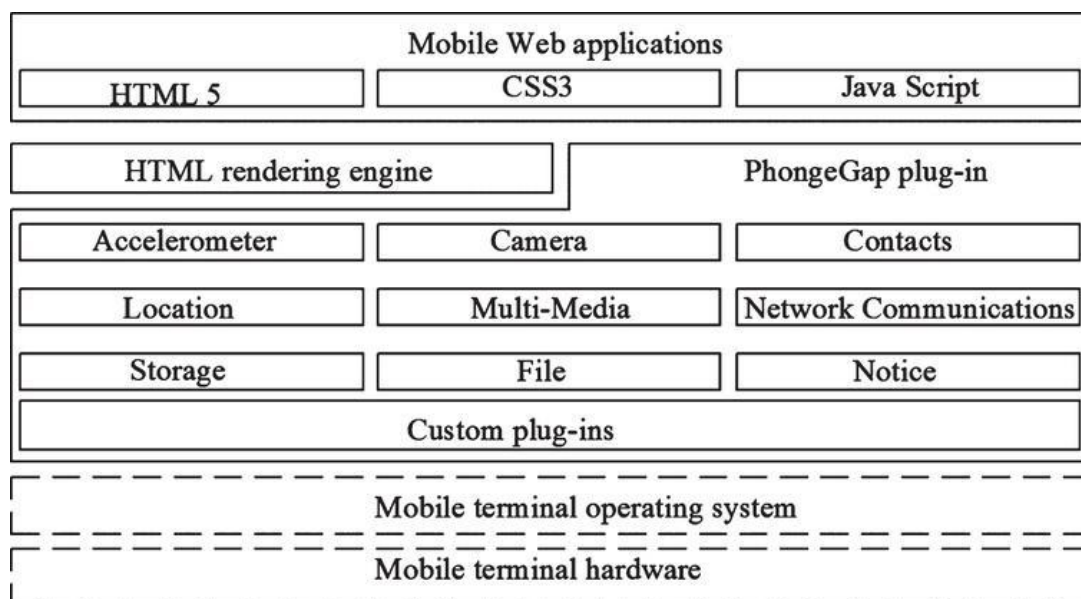


HTML5 标准制定完成后，受到了移动互联网开发者的广泛关注。在 HTML5 标准中，为了解决网络在移动互联网应用中问题，HTML5 标准提供了许多新的技术特性。这些技术特性包括本地存储、图形、多媒体回放、网络通信以及地理位置和移动网络框架的使用。这些新功能使移动 Web 应用程序能够实现各种丰富内容的网络应用程序，如图形、移动设备、离线缓存以及 PC 平台上的浏览器视频回放，从而实现功能多样化。此外，HTML5 标准克服了浏览器过于依赖插件的问题，使移动网络设备能够在有限的内存空间中提供更多便利的条件。此外，在手机、pad 等主流移动操作系统中，浏览器使用的 webkit 内核可以支持 html5 标准和 css3 标准，使相同的应用软件可以跨平台使用。它具有丰富的功能，由于 HTML5 的新特点和功能节省了存储资源，且跨平台，使得应用程序版本方便控制，所以，移动互联网应用程序将成为未来应用的主流。

移动互联网中的各种 Web 应用程序在很大程度上依赖于硬件以及操作系统。目前市场上主要有三种操作系统，即 Android、iOS 和 Windows。过去，硬件操作系统的不同，使得移动 web 应用开发人员必须在同一类型操作系统上进行开发，以保证软件群体的使用，应用软件的开发和维护成本迅速上升。随着 HTML5 标准的发布和大规模应用，可跨平台应用的移动 web 应用软件已经广泛应用于移动互联网，如百度地图移动版、爱奇艺、优酷等。他们都使用 HTML5 的新功能来实现音频和视频回放。这些基于 HTML5 标准开发的 Web 应用程序在使用性能和用户体验方面具有很好的体验。各大公司逐渐放弃了对移动设备上 Flash 插件的支持，研究了各种基于 HTML5 标准的跨平台应用，使得 HTML5 在移动互联网上的应用更加广泛。然而，基于 HTML5 标准，移动设备的硬件需要通过操作系统浏览器的 API 来实现适配移动 Web 应用软件。因此，浏览器需要具有良好的 HTML5 支持功能。但是，当前主流操作系统的浏览器对 HTML5 标准有不同的支持度，使得

HTML5 不能在某些浏览器中使用。

目前，HTML5 在移动互联网上的主流应用是混合应用（hybrid app），这是为了解决一些浏览器不能支持 HTML5 标准的问题。所谓的混合应用程序是指移动 Web 应用程序和操作系统应用程序，它们可以在各自的应用程序之间进行折衷。通过结合 HTML5 标准的新功能和移动 web 应用的需求，混合应用（hybrid）具有移动 web 应用和操作系统应用的优势。通过混合方法的应用，软件开发人员利用 HTML5 技术克服了系统自身的软件平台和碎片化问题。通过 HTML5 标准中 JavaScript 技术的混合应用插件，可以兼容不能完全支持设备 API 问题的浏览器，从而充分发挥 HTML5 标准的各种特性。目前市场上混合应用的开发框架很多，目前主流的开发框架有 PhoneGap、AppMobi、AppCan 和 Titanium。除了混合应用之外，移动互联网中的 HTML5 应用还包括移动 web 应用和 web 应用商店，其中移动 web 应用在兼容 HTML5、跨平台和开发成本方面不如混合应用好。



附录 源程序

后台 SqlMap 部分

```
//
sql
语句

var sqlMap = {
    // 用户
    leader: {
        query: 'select * from account where sphone = ? and spass = ?',
        query_dic: 'select * from dictionary order by did desc',
        add_dic: 'insert into dictionary(dxynome,dxiname,dzname,dnjname,dbjname)
value(?,?,?,?,?)'
    },
    teacher: {
        query: 'select * from teacher where tphone = ? and tpassword = ?',
        class: 'select d.did , d.dnjname,d.dzname,d.dbjname from teacher t,dictionary
d,teacherclass tc where t.tid = ? and tc.dictionaryid = d.did and tc.teacherid
= t.tid',
        record_query: 'select s.sname,s.ssex,s.sphone,
q.qdate,(q.qdate+qtime)>(q.qdate+qstarttime) ss ,q.qteachermsg'
        +' from qdrecord q,student s where q.studentno = s.sno and s.classno= ?',
        query_qingjia: 'select sname,qtype,qres,qdate from qingjiarecord'
    },
    student: {
        query: 'select * from student where sphone = ? and spassword = ?',
        record_qd: 'INSERT INTO
qdrecord(studentno,qdate,qtime,qstarttime,qteachermsg) VALUES (?,?,?,?,?)',
        querybyname: 'select s.sname,s.ssex,s.sphone, q.qdate,q.qteachermsg'
        +' from qdrecord q,student s where q.studentno = s.sno and s.sno= ?',
        // 'select * from qdrecord where studentno = ?'
        qingjia: 'INSERT INTO
qingjiarecord(studentno,sname,qtype,qres,qdate,teacherid) VALUES
(?,?,?,?,?,34)',
    },
    echart: {
        echart_time: ' select count(q.qdate) as times from qdrecord q where q.qdate
like ? ' +
        'union all select count(q.qdate) as times from qdrecord q where q.qdate like ?'
        +
        'union all select count(q.qdate) as times from qdrecord q where q.qdate like ?'
```

```

+
    'union all select count(q.qdate) as times from qdrecord q where q.qdate like ?'
+
    'union all select count(q.qdate) as times from qdrecord q where q.qdate
like ?' ,
    echart_people: 'select count(q.qteachermsg) as nn from qdrecord q where
q.qteachermsg = ?' +
    'union all select count(r.qteachermsg) as nn from qdrecord r where r.qteachermsg
like ?',
    }
}

module.exports = sqlMap;

```

后台签到实现部分

```

var express = require('express');
var router = express.Router();
var mysql = require('mysql');
var $sql = require('../sqlMap');

// 连接数据库
var conn = mysql.createConnection(models.mysql);

conn.connect();
var jsonWrite = function(res, ret) {
    if(typeof ret === 'undefined') {
        res.json({
            code: '1',
            msg: '操作失败'
        });
    } else {
        res.json(ret);
    }
};

//[params.studentno,params.qdate,params.qtime,params.qstarttime]
//[params.classno]
// 进行签到
router.post('/recordQd', (req, res) => {
    var sql = $sql.student.record_qd;
    var params = req.body; //传入四个数据,
    console.log(params);
    conn.query(sql,

```

```
[params.studentno,params.qdate,params.qtime,params.qstarttime,params.qteachermsg], function(err, result) {
    if (err) {
        console.log(err);
    }
    if (result) {
        jsonWrite(res, result);
    }
})
});

module.exports = router;
```

前端发布二维码部分

```
//SVG 可缩放矢量图形 (Scalable Vector Graphics) 绘制二维码, 按照班级
createQr(cid,url,time){
    const qrcode = new QRCode(document.querySelector('#qrcode'), {
        //text: 'your content',
        width: 100,
        height: 100,
        colorDark: '#000000',
        colorLight: '#ffffff',
        correctLevel: QRCode.CorrectLevel.H, //容错级别
        useSVG: true
    })
    qrcode.makeCode(url+'?did='+cid+'&time='+time)
},
// 放大或者缩小
zoom(num) {
    var svg = document.querySelector('#svg')
    this.scale = this.scale * num
    if (this.scale < 0.5 || this.scale > 1.1) {
        return
    } else {
        svg.setAttribute('transform', 'scale(' + this.scale + ')')
    }
},
//当选择完后调用 qrcode, 绘制二维码
select(){
    let cid = this.listVaule.slice(0,2)//匹配开始到编的数据
    if(this.startTime){
        this.$options.methods.createQr(cid,this.qrUrl,this.startTime)//
```

```

methods 中调用另外一个声明函数要使用 this.$options.methods.函数名()
    }
  },

```

移动端签到实现

//麻烦点在于时间格式的转化，要做对比，这里可以后台做对比也可以前台做对比，主要是签到状态

//mysql 里的 datetime 不能存 JavaScript 的默认时间值，

//两种方法，把 datetime 改为 text 缺点不太知道，还有就是做拼接，拼成 datetime 的样子

//签到功能，相当于一次，获取扫描二维码后传参地址里面的关键值 cid，存储的时候要记录当前时间，

```

qd() {
  let data = {
    'studentno': '', //学生编号
    'qdate': '', //签到的年月日
    'qtime': '', // 签到时分秒
    'qstarttime': '', //最迟签到时间
    'qteachermsg': '', //是否迟到-----》前端实现和后端实现
  }
  var nowDate = new Date() //获取现在时间
  let sno = JSON.parse(window.localStorage.getItem('studentInfo')).sno //
  获取 webStorage 中的学生 id
  let qstarttime = GetQueryString('time') // 获取 hash 中的 time 值 这个 time 值
  就是教师设定的签到时间，在这之前为正常，之后为迟到
  if (qstarttime != null && qstarttime.toString().length > 1) {
    //赋值
    data.studentno = sno
    data.qdate = toDateTime(nowDate) // 转换现在时间
    -----需要处理时间 toDateTime

    data.qstarttime = qstarttime
    data.qtime = nowDate.getHours() + ':' + nowDate.getMinutes() + ':' +
    nowDate.getSeconds()
    //data.qteachermsg 需要解析时间
    let qstartHour = qstarttime.substr(0,2)
    let qstartMinutes = qstarttime.substr(3,2)
    if(nowDate.getHours() < qstartHour){
      data.qteachermsg='正常'
    }
    else if(nowDate.getHours() >= qstartHour)
    {
      if(nowDate.getMinutes() > qstartMinutes){

```

```
        let lateTime = nowDate.getMinutes() - qstartMinutes
        data.qteachermsg = '迟到'+lateTime+'分钟'
      }else{
        data.qteachermsg = '掐点到'
      }
    }
    console.log(data)
    this.$axios.post('/api/record/recordQd', data).then((response) => {
      console.log(response)
      if (response.statusText === 'OK') {
        this.$message({
          type: 'success',
          message: '签到成功'
        })
      } else {
        this.$message.error('签到失败')
        this.err()
      }
    })
  }
},
```