基于安卓的定制化数字课堂考勤考评系统

参赛学校：上海海洋大学

参赛成员：蔡佳昊 孙翊佳 马耀

2017-12-24

目录

[目录 1](#_Toc502839545)

[1.简介 3](#_Toc502839546)

[1.1作品创意/项目背景 3](#_Toc502839547)

[1.2项目实施计划 4](#_Toc502839548)

[1.2.1 人员分工 4](#_Toc502839549)

[1.2.2 进度安排 4](#_Toc502839550)

[2.总体设计 5](#_Toc502839551)

[2.1系统功能 5](#_Toc502839552)

[2.1.1功能概述 5](#_Toc502839553)

[2.1.2功能说明 5](#_Toc502839554)

[2.2系统软硬件平台 6](#_Toc502839555)

[2.2.1系统开发平台 6](#_Toc502839556)

[2.2.2系统运行平台 7](#_Toc502839557)

[2.3关键技术 8](#_Toc502839558)

[2.3.1Android 8](#_Toc502839559)

[2.3.2Servlet 8](#_Toc502839560)

[2.3.3MySQL 8](#_Toc502839561)

[2.3.4JSON 9](#_Toc502839562)

[2.4作品特色 9](#_Toc502839563)

[2.4.1课堂教室数字化 9](#_Toc502839564)

[2.4.2将电子栅栏技术用于课堂考勤，剔除作弊签到情况 9](#_Toc502839565)

[2.4.3将设备唯一标识码用于课堂考勤，剔除作弊代签情况 9](#_Toc502839566)

[2.4.4基于数字化课堂的实时点评 10](#_Toc502839567)

[2.4.5净化课堂环境 10](#_Toc502839568)

[2.4.6学生数据Excel导出 10](#_Toc502839569)

[2.5作品效果 10](#_Toc502839570)

[2.5.1提速考勤流程 10](#_Toc502839571)

[2.5.2提速考评/信息维护流程 10](#_Toc502839572)

[2.5.3加强学生课堂参与度和互动积极性 11](#_Toc502839573)

[2.5.4提升课堂教学效率和质量 11](#_Toc502839574)

[3.详细设计 12](#_Toc502839575)

[3.1系统结构设计 12](#_Toc502839576)

[3.1.1技术架构 12](#_Toc502839577)

[3.1.2功能模块设计 12](#_Toc502839578)

[3.2.1存储结构设计 12](#_Toc502839579)

[3.3活动（Activity）设计与实现 17](#_Toc502839580)

[3.3.1 整体UI界面设计风格 17](#_Toc502839581)

[3.3.2教师用户组活动 17](#_Toc502839582)

[3.3.3学生用户组活动 23](#_Toc502839583)

[3.3.4公共用户组活动 26](#_Toc502839584)

[3.4服务端结构设计与实现 33](#_Toc502839585)

[3.4.1数据库的增删改查 34](#_Toc502839586)

[4总结 36](#_Toc502839587)

[4.1孙翊佳谈作品的创意 36](#_Toc502839588)

[4.2蔡佳昊谈开发实现过程 36](#_Toc502839589)

[4.3马耀谈团队合作心得体会 37](#_Toc502839590)

[三人谈作品的改进方向 38](#_Toc502839591)

[5附录 39](#_Toc502839592)

[5.1参考资料 39](#_Toc502839593)

[5.1.1书籍 39](#_Toc502839594)

[5.1.2博客 39](#_Toc502839595)

[5.1.3专业术语 40](#_Toc502839596)

[5.2源代码清单 43](#_Toc502839597)

1.简介

1.1作品创意/项目背景

随着高等教育改革的推进深入，国内高等院校录取人数逐年增加，在2017年全国本科招生计划372万人，真正实现了“高等教育普及化”的改革目标，但是随之而来的相关问题也困扰着高校师生。由于高校扩招，部分教师的教学班级增多，部分教学班级的上课学生人数增多，以至于课堂教师对学生的考勤考评工作量变大。另外，由于教室里的学生人数增多，上课学生的签到，以及任课教师的点名总是会占据大量的上课时间，大大的降低了有效的教授知识的时间。市场上存在的 “蓝墨云”和“好班”App，试图解决现状，但却存在过于商业化和娱乐化的弊端。在下文中将详细阐述现存的问题。

第一个问题是:教师教授的学生数量增多，从而增加教师对学生的考勤考评的工作量增加。根据团队在学校的调查研究表明，教师在60人的教室里进行点名记录每次需要花费5分钟的时间，每天按两次的点名次数计算，教师就会失去10分钟的有效授课时间。有效授课时间的减少势必降低教学效。

其次，学生上课的部分学习时间被点名签到时间占用。在老师进行点名的时候，学生们的注意力通常都在点名这件事上，很难集中精力的把心思放在学习上面。每次点名需要花费5分钟，这使得原本45分钟的上课时间被缩减掉11%，日积月累，这将浪费学生大量的学习时间。

第三个问题:学生数量的增加也给辅导员的工作带来了挑战，辅导员很难了解到真正需要教育指导的学生。辅导员的职责主要是为学生提供帮助，尤其是有问题的学生。然而，随着学生数量的增加，辅导员很难去了解每一位学生。再者，过去辅导员了解学生的情况通常是结合学生的平常考评情况以及期末总评，而这些数据通常只会在每学期期末的时候教务处才予以更新，这就是说辅导员拿到数据的时候已经错过了对学习态度不端正的学生的最佳的教育时间。

第四个问题:市场上存在的为解决点名考评的“蓝墨云”和“好班”App过分的商业化和娱乐化，而且没有教室座位号的信息笼统而不精准。“蓝墨云”和“好班”App虽然实现了教师可以在线上对学生进行点名考评，但是他们接入了广告界面、电商页面以及第三方的流量入口，诱导学生流入与课堂学习无关的其他网络网站。娱乐化也使得部分学生在课上使用其与其他人进行社交沟通从而耽误上课时间。

总的来说，现今高校的考勤考评方式普遍存在效率低下、浪费有效课堂教学时间以及辅导员难以第一时间获取学生考评信息等问题，而且市面上的关于高校考勤考评的App也存在对各高校教室的座位号缺乏调研、以及过度商业化和娱乐化的通病。综合上文中提到的在教学考评中存在的种种问题，我们团队拟研究开发一款绿色且轻便的基于移动端的定制化数字课堂考勤考评系。

1.2项目实施计划

1.2.1 人员分工

蔡佳昊：客户端和服务端编码实现

马耀：数据库设计与实现

孙翊佳：立项，需求分析，界面设计

1.2.2 进度安排

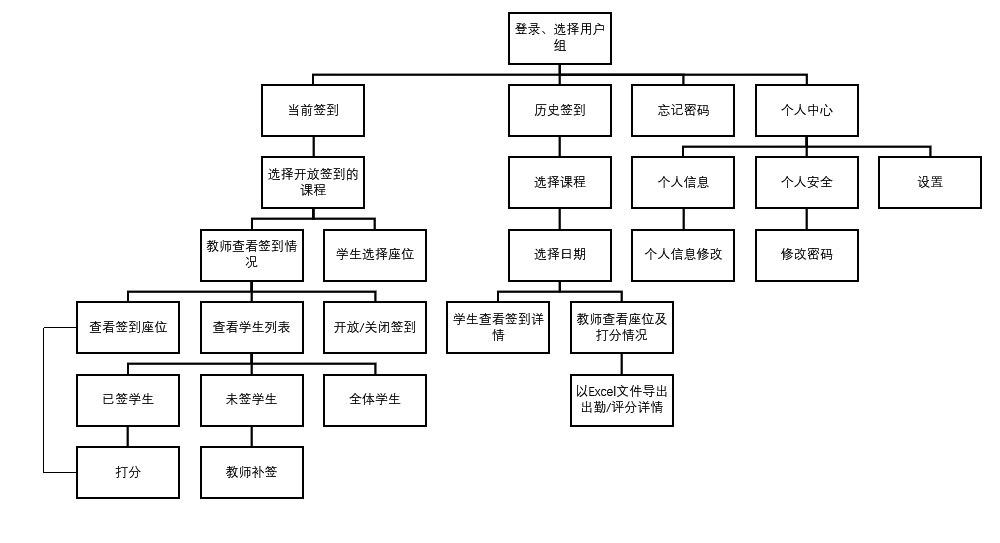
|  |  |  |
| --- | --- | --- |
| **日期** | **工作** | **备注** |
| 2017.3 | 提出项目构思 |  |
| 2017.5 | 确定技术方向，确定核心需求 |  |
| 2017.5-2017.9 | 学习开发必要的知识、技术。  完成总体设计 |  |
| 2017.9-2017.12 | 编码 |  |
| 2017.12-2018.1 | 撰写软件文档 |  |
| 2018.1 | 模拟环境仿真测试 |  |
| 2018.1-2018.3 | 完善逻辑功能、UI、界面和文档 |  |

2.总体设计

【本实现的功能、开发技术和应用环境进行介绍。】

2.1系统功能

2.1.1功能概述

系统主要应用于教师对学生进行考勤、考评等课堂日常功能需求。详细功能分布安排如下图1所示。

**图1：iClass整体功能分布图**

2.1.2功能说明

基于Android的课堂考勤考评服务App是基于Android开发技术展开，主要面向师生群体，在用户层面上分成2个用户组：教师和学生，主要分为教师课堂考勤和学生课堂签到，教师课堂考评，学生信息查询和导出等功能群。

2.1.2.1教师课堂考勤和学生课堂签到

教师在课堂上口头告诉学生即将开始签到，教师一键开始签到，学生选择好自己要坐的座位后即完成签到。老师可以选择一定时间后关闭本次签到，若无法再使用手机也可以课下再关闭，晚签到的同学都会有对应迟到信息。教师也可以用自己的手机来帮助由于手机问题无法完成签到的学生完成签到。

2.1.2.2教师课堂考评

教师在教师端上点击座位图中的座位或者在已签学生列表中选择学生，即能给指定的这个学生进行考评；也可以打开当前教学班级学生名单对某一学生进行考评操作。

2.1.2.2学生信息查询和导出

教师在教室端上选择需要查询的教学班级信息，再选择具体上课节数和时间，即能调取教学班级学生历史记录信息，查看学生过去的出勤、上课考评信息，导出Excel文件操作。

2.2系统软硬件平台

2.2.1系统开发平台

2.2.1.1操作系统

名称：Windows 10

开发公司：Microsoft

版本号:：最新版本

2.2.1.2开发平台

名称：Android Studio

开发公司：Google

版本号：2.3.2

名称：Eclipse IDE for Java EE Developers

开发公司：Oracle

版本号：Oxygen.1a Release (4.7.1a)

2.2.1.3数据库

名称：MySQL

开发公司：MySQL AB

版本号：5.7

2.2.1.4数据库可视化工具

名称：MySQL Workbench

开发公司：MySQL AB

版本号：6.3.9

名称：Navicat for MySQL

开发公司：PremiumSoft

版本号：10.1.7

2.2.1.5测试服务器

名称：Apache Tomcat

开发公司：Apache

版本号：7.0.82

平台：Windows10

2.2.1.6版本控制系统

名称：git

版本号：2.15.1.2-64-bit

平台：Windows10

名称：github

项目网站：

<https://github.com/spencercjh/iClass>

https://github.com/spencercjh/iClass\_Sever

2.2.2系统运行平台

2.2.2.1手机端配置要求：

Android手机

屏幕大小：5.0英寸

系统要求：Android 5.0以上

2.3关键技术

2.3.1Android

Android是一种基于Linux的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由Google公司和开放手机联盟领导及开发。尚未有统一中文名称，中国大陆地区较多人使用“安卓”或“安致”。Android操作系统最初由Andy Rubin开发，主要支持手机。2005年8月由Google收购注资。2007年11月，Google与84家硬件制造商、软件开发商及电信营运商组建开放手机联盟共同研发改良Android系统。随后Google以Apache开源许可证的授权方式，发布了Android的源代码。第一部Android智能手机发布于2008年10月。Android逐渐扩展到平板电脑及其他领域上，如电视、数码相机、游戏机等。2011年第一季度，Android在全球的市场份额首次超过塞班系统，跃居全球第一。 2013年的第四季度，Android平台手机的全球市场份额已经达到78.1%。2013年09月24日Google开发的操作系统Android在迎来了5岁生日，全世界采用这款系统的设备数量已经达到10亿台。

2.3.2Servlet

Servlet（Server Applet）是Java Servlet的简称，称为小服务程序或服务连接器，用Java编写的服务器端程序，主要功能在于交互式地浏览和修改数据，生成动态Web内容。

狭义的Servlet是指Java语言实现的一个接口，广义的Servlet是指任何实现了这个Servlet接口的类，一般情况下，人们将Servlet理解为后者。Servlet运行于支持Java的应用服务器中。从原理上讲，Servlet可以响应任何类型的请求，但绝大多数情况下Servlet只用来扩展基于HTTP协议的Web服务器。

最早支持Servlet标准的是JavaSoft的Java Web Server，此后，一些其它的基于Java的Web服务器开始支持标准的Servlet。

2.3.3MySQL

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

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

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

由于其社区版的性能卓越，搭配 PHP 和 Apache 可组成良好的开发环境。

2.3.4JSON

JSON(JavaScript Object Notation, JS 对象标记) 是一种轻量级的数据交换格式。它基于 ECMAScript (w3c制定的js规范)的一个子集，采用完全独立于编程语言的文本格式来存储和表示数据。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言。 易于人阅读和编写，同时也易于机器解析和生成，并有效地提升网络传输效率。

2.4作品特色

2.4.1课堂教室数字化

紧密结合学校教学楼座位分布情况，实现各个教室座位无差别地呈现于手机屏幕上，方便学生考勤选座和教师课间评分。

2.4.2将电子栅栏技术用于课堂考勤，剔除作弊签到情况

为教学楼建立平面几何图形模型，学生点选教室座位图中的座位后，系统会获取学生当前的位置信息，结合学生经纬度定位坐标，判断学生位置，防范恶劣作弊签到。

2.4.3将设备唯一标识码用于课堂考勤，剔除作弊代签情况

将登陆过的学生和教师用户的设备唯一标识码上传至数据库，若用户再用此设备登陆其他账号，即给出警告信息，不允许继续登陆和进一步操作，防范恶劣作弊代签。

2.4.4基于数字化课堂的实时点评

教师可以通过本项目的预期成果给予先前课上给其留下深刻印象的学生好评或者是差评，以此调动学生课堂学习的积极性。

2.4.5净化课堂环境

教师/学生在开启/完成签到后手机将自动被设置为静音，让课堂免于铃声困扰。用户可以在设置中开启/关闭这个功能。

2.4.6学生数据Excel导出

教师在历史签到中可选择某一个教学班并导出签到及考评详细情况。

2.5作品效果

使用本项目的预期成果后，辅导员、老师和学生之间将形成“三位一体”的紧密联系，最大限度地减少教师点名的时间，最大限度地提升课堂的教学速率和质量，最大限度地提升学生出勤率。本项目的预期成果区别于市面上其他的考勤App，功能上提供绿色的考勤、考评服务，通过简单的参数修改将拓展出适用于各高校且富有各高校特色的定制化数字课堂考勤考评系统。再通过用户的体验分享，口碑传播，日后势必将成为中国各高校的新型考评方式。

2.5.1提速考勤流程

使用本项目的预期成果，教师能够随时发动签到。本项目的预期成果将原本串行的点名流程替换成并行过程。无论学生数量的多少，都能够同时在几秒钟内完成考勤、签到的过程。

2.5.2提速考评/信息维护流程

使用本项目的预期成果，教师能够摆脱繁杂冗余的学生名册，能清楚直观地查看已到、未到、全体学生列表。本项目的预期成果的特色在于不仅将学生名表数字化，还将教室座位数字化，教师也能清楚直观地查看教室座位和学生的落位情况，能够快速对某个座位上的某个学生进行加分或扣分等信息维护操作。

2.5.3加强学生课堂参与度和互动积极性

使用本项目的预期成果，教师能通过关闭后排座位强制让学生坐在教室前排。教师的快速评分，能够让学生更多地参与进课堂活动中来，更能让学生的课堂表现与分数、评价真正挂钩起来。

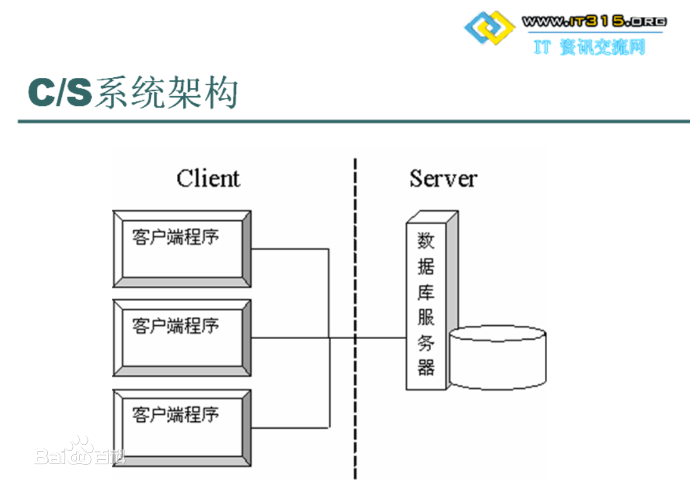
2.5.4提升课堂教学效率和质量

使用本项目的预期成果后，考勤点评的耗时大大压缩，课堂有效教学时间大大提高。通过准确严格的考勤流程去约束学生，能够提升课堂的出勤率，减少早退、迟到情况的出现。实现快速考评后，学生课堂参与度提高，课堂氛围更为活跃，课堂教学质量就能稳步上升，良好学风也能逐步建立。3详细设计与实现

3.详细设计

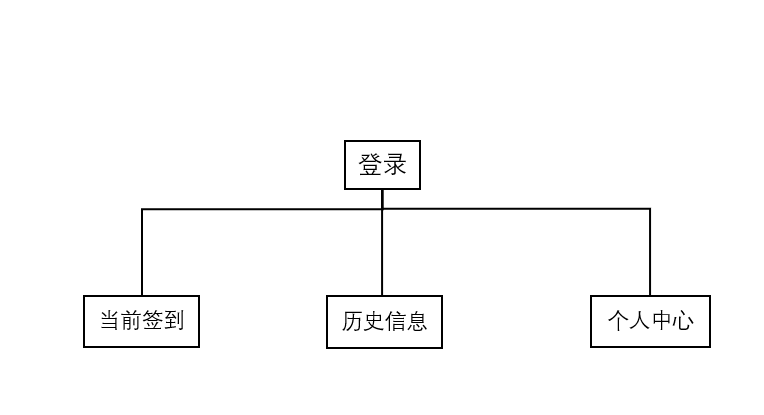
3.1系统结构设计

3.1.1技术架构

系统整体架构采用C/S架构，基于客户端/服务器(C/S)模式的结构将客户端与数据库相结合，形成的基于数据库的客户端计算模式，并将该模型应用到Internet/Intranet中。 如下图所示：

**图2：iClass系统架构图**

3.1.2功能模块设计

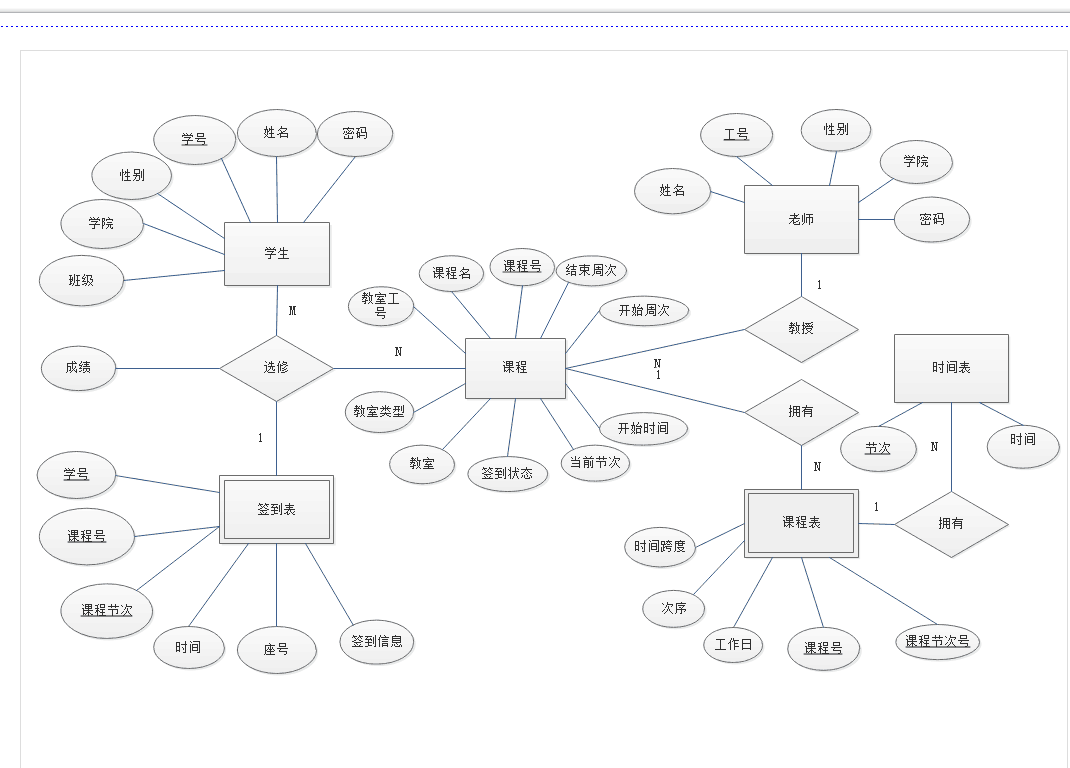
整体设计中依据业务流程的环节以及特点分为教师和学生的课堂活动，教师的课下学生历史信息调阅、用户个人资料维护三大功能群提供服务。大部分功能逻辑使用Android Studio在活动(Activity)Java文件中实现，小部分功能逻辑在服务端中Servlet的Java文件中实现。如下图所示：

**图3：iClass功能模块划分图**

3.2.1存储结构设计

使用MySQL Workbench数据库可视化工具，从数据关系、数据字典等方面着手进行数据库的存储结构设计与实现。App中涉及到的大部分数据（除设置中的配置文件）都存储在服务端的MySQL服务器中。

3.2.1.1数据库ER图：



**图4：iClass服务端数据库ER图**

3.2.1.2数据字典：

服务端数据库数据字典如表6至表所示：

**表1 签到信息表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据表名:all\_check\_info 中文描述：签到信息表 | | | | | | |
| 字段名称 | 字段描述 | 数据类型 | 长度 | 是否允许空 | 缺省值 | 备注 |
| subject\_id | 课程编号 | Char | 15 | 否 |  |  |
| subject\_th | 课程已至节数 | Int |  | 否 |  |  |
| student\_id | 学生 | Char | 15 | 否 |  |  |
| seat\_index | 座位号 | Int |  | 否 |  |  |
| ischeck | 主管部门 | Int |  | 否 |  |  |
| time | 签到时间 | DATETIME | 6 | 否 |  |  |

**表2 学生信息表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据表名:student 中文描述：学生信息表 | | | | | | |
| 字段名称 | 字段描述 | 数据类型 | 长度 | 是否允许空 | 缺省值 | 备注 |
| id | 学生学号 | Char | 15 | 否 |  |  |
| name | 学生姓名 | Char | 15 | 否 |  |  |
| password | 学生密码 | Char | 15 | 否 |  |  |
| sex | 学生性别 | Int |  | 是 | NULL |  |
| college | 学生学院 | Char | 15 | 是 | NULL |  |
| class | 学生班级 | Char | 15 | 是 | NULL |  |

**表3 教师信息表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据表名:teacher 中文描述：教师信息表 | | | | | | |
| 字段名称 | 字段描述 | 数据类型 | 长度 | 是否允许空 | 缺省值 | 备注 |
| id | 教师工号 | Char | 15 | 否 |  |  |
| name | 教师姓名 | Char | 15 | 否 |  |  |
| Password | 教师密码 | Char | 15 | 否 |  |  |
| Sex | 性别 | Int |  | 否 |  |  |
| College | 院系 | Char | 15 | 否 |  |  |

**表4 课程信息表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据表名:subject 中文描述：课程信息表 | | | | | | |
| 字段名称 | 字段描述 | 数据类型 | 长度 | 是否允许空 | 缺省值 | 备注 |
| id | 课程编号 | Char | 15 | 否 |  |  |
| name | 课程名 | Char | 15 | 否 |  |  |
| Section | 课程已至节数 | Int |  | 否 |  |  |
| teacher\_id | 任课老师工号 | Char | 15 | 否 |  |  |
| class\_type | 教师类型 | Int |  | 否 |  |  |
| classroom | 教室位置 | Char | 15 | 否 |  |  |
| check\_situation | 签到状态 | Int |  | 否 | 0 |  |
| start\_time | 开始签到时间 | Char | 15 | 是 | NULL |  |
| Begin\_Week | 开始周次 | Int |  | 否 | 1 |  |
| End\_Week | 结束周次 | Int |  | 否 | 16 |  |

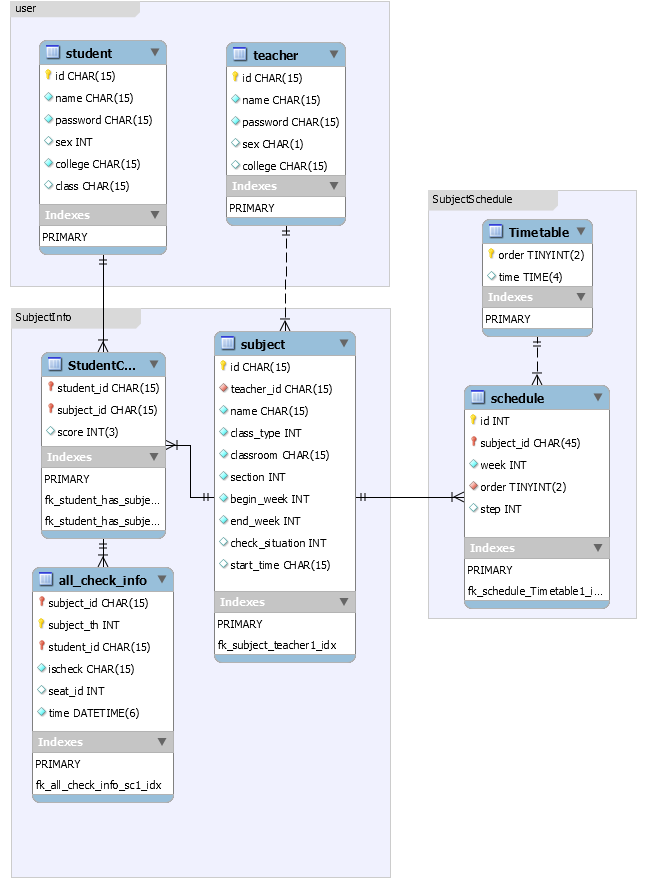
**表5 学生课程信息表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据表名:StudentCourse 中文描述：学生课程信息表 | | | | | | |
| 字段名称 | 字段描述 | 数据类型 | 长度 | 是否允许空 | 缺省值 | 备注 |
| subject\_id | 课程编号 | Char | 15 | 否 |  |  |
| student\_id | 学生编号 | Char | 15 | 否 |  |  |
| Score | 课程成绩 | Int |  | 是 | Null |  |

**表6 课程-学生名表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据表名:subjectSchedule 中文描述：课程-时间表 | | | | | | |
| 字段名称 | 字段描述 | 数据类型 | 长度 | 是否允许空 | 缺省值 | 备注 |
| Subject\_id | 课程编号 | Char | 15 | 否 |  |  |
| Id | 课程节次编号 | Char | 15 | 否 |  |  |
| Week | 上课的工作日 | Int |  | 否 |  |  |
| Order | 课程节次 | Int |  | 否 |  |  |
| Step | 课程上课时间跨度（周） | Int |  | 否 | 1 |  |

3.2.1.3数据表关系：



**图5：iClass服务端数据表关系图**

3.3活动（Activity）设计与实现

包括三大部分，教师用户组主要活动、学生用户组主要活动和公共活动。大部分活动的界面使用Android Studio在App的布局(Layout)xml文件中实现，小部分按键、动画效果需要在活动(Activity)Java文件中实现。一些活动采用一个活动Activity文件对应多个布局(Layout)文件的实现方式以区分用户组的不同。

3.3.1 整体UI界面设计风格

如上图所示，本App的UI界面以天蓝色底色为主，象征着“天空”，象征着当代大学生自由地在知识的天空翱翔。天蓝色也符合大部分用户的心理需要，给人以大方、有序之感；以深蓝色为工具栏底色，展现上海海洋大学特色。部分核心功能页面采用导航式菜单布局，方便用户快速进入功能模块的同时降低用户操作深度，取得较好的用户体验。

3.3.2教师用户组活动

3.3.2.1选择需要签到的课程界面（fragment）

登录活动会向这个活动传一个字符串参数“user”，若它是“teacher”即为教师用户组则显示以下界面。

向服务器发起请求，查询当前教师用户执教的课程，利用listview组件按照“@课程名@课程编号@教室位置@学生人数@签到情况”的条目顺序显示。

单击某一门课程，进入那门课程的签到情况页面。

工具栏利用toolbar组件，上有刷新键和溢出菜单栏，内有退出登录和添加课程按键（但添加课程并不对教师开放）。

点击刷新建，向服务器发起请求，查询当前教师用户执教的课程，利用listview组件按照“@课程名@课程编号@教室位置@学生人数@签到情况”的条目顺序显示。

点击退出登录按键，进入登录活动，但会传输一个字符串参数“exit”，进入登陆活动后接收到字符串参数后会把已经填写过的表单信息清除掉。

点击添加课程按键，进入添加课程活动，但我们并没有允许教师自行添加课程，这是由后台管理员添加的。这个活动是为学生用户组而做的。

界面如下图所示：

  
**图7：选择需要签到的课程界面**

3.3.2.2查看点选的课程签到情况活动

向服务器发出请求，查询表subject中的studentnum即是这门课应到人数，查询all\_check\_info中属性subject\_id和subject\_th与这节课对应的信息条数即是这门课实到人数，利用2个Textview组件显示2个数字。

工具栏利用toolbar组件，上有倒退一次上课节数按钮，开始签到按钮，刷新按钮和返回键。

点击倒退一次上课节数按钮，向服务器发出请求，将表subject中这门课程对应的属性subject\_th减一，并提供限制逻辑，减到0时不再能减下去，并将这门课对应的属性check\_situation改为0即关闭签到。

点击开始签到按钮，向服务器发出请求，将表subject中这门课程对应的属性subject\_th加一，并将这门课对应的属性check\_situation改为1即开放签到，并将按钮文字改成“结束签到！”。

点击刷新按钮，向服务器发出请求，查询表subject中的studentnum即是这门课应到人数，查询all\_check\_info中属性subject\_id和subject\_th与这节课对应的信息条数即是这门课实到人数，利用2个Textview组件显示2个数字。

点击查看学生列表按钮，进入查看学生列表活动。

点击查看签到座位图按钮，进入查看签到座位图界面。

点击结束本次签到按钮，向服务器发出请求，将这门课对应的check\_situation属性改为0即关闭签到，将按钮文字改成“开始签到！”

点击返回键，关闭这个页面，回到选择需要签到的课程活动。

如下图所示：

**图8：查看点选的课程签到情况活动界面**

3.3.2.3查看学生列表活动

工具栏利用toolbar组件，上有已签，未签，全体和返回键。

点击已签按钮，将工具栏下面的界面替换成已签学生列表界面。向服务器发出请求，查询表all\_check\_info中属性subject\_id和subject\_th与这节课对应的签到信息，使用json转换，恢复ArrayList数据结构，这个ArrayList存放着这个班这节课的已签学生。利用listview组件按照“@姓名@学号@签到时间@签到状态”的条目顺序显示。单击一个学生条目，进入查看学生详细资料/打分活动。

点击未签按钮，将工具栏下面的界面替换成未签学生列表界面。向服务器发出请求，查询表subject\_xxxxxxx（课程编码）中的学生信息，使用json转换，恢复ArrayList数据结构，这个ArrayList存放着这个班的全体学生。向服务器发出请求，查询表all\_check\_info中属性subject\_id和subject\_th与这节课对应的签到信息，使用json转换，恢复ArrayList数据结构，这个ArrayList存放着这个班的已签学生。双层循环，将全体学生中的已签学生以外的学生信息条目加入新ArrayList，这个ArrayList就存放着这个班的未签学生。利用listview组件按照“@姓名@学号@学院@班级@出勤@评分”的条目顺序显示。单击一个未签学生条目，弹出警告框，询问教师是否要为这名学生代签。

点击全体按钮，将工具栏下面的界面替换成全体学生列表界面。向服务器发出请求，查询表subject\_xxxxxxx（课程编码）中的学生信息，使用json转换，恢复ArrayList数据结构，这个ArrayList存放着这个班级的全体学生。利用listview组件按照“@姓名@学号@学院@班级@出勤@评分”的条目顺序显示。

这个活动默认显示已签学生列表界面。

点击返回键，关闭这个页面，回到查看点选的课程签到情况活动。

界面如下图所示：

**图9：查看学生列表活动界面**

3.3.2.4查看签到座位图活动

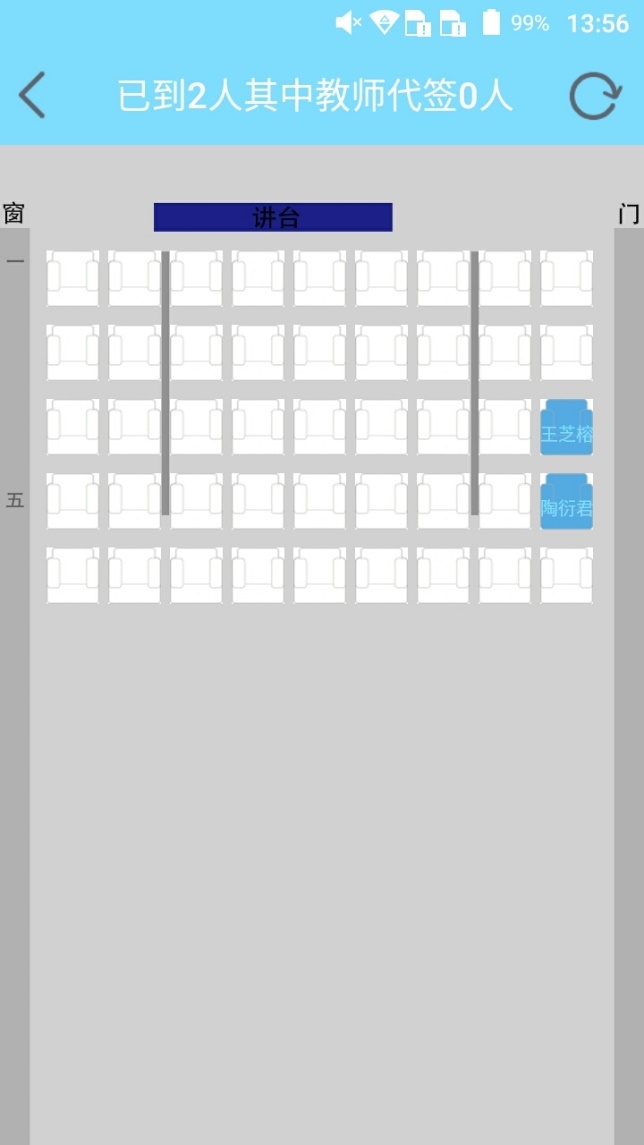
工具栏利用toolbar组件，toolbar的title即时显示已签多少学生：访问服务器，查询all\_check\_info中属性subject\_id和subject\_th与这节课对应的信息条数即是这门课实到人数。工具栏上还有返回和刷新键。

工具栏下面的界面利用gridview组件显示了一个座位图。显示之前先访问服务器，查询表subject中这节课程对应的属性class\_type，根据class\_type显示小、中、大、特大尺寸的教室对应的座位图；查询表all\_check\_info中属性subject\_id和subject\_th与这节课对应的签到信息，使用json转换，恢复ArrayList数据结构，这个ArrayList存放着这个班的已签学生；创建一个大小为教室座位总数的布尔数组，以学生的座位号seat\_index为下标，若有人坐了就设置table[seat\_index]为true;再创建一个大小为教室座位总数的签到信息类数组，将座位图先抽象为一个数组，哪里坐了学生就下标对应的元素是那个学生的签到信息。这样一来，在构建座位图的循环中就不会有访问服务器请求，极大降低了时间成本。在显示座位图的时候若对应seat\_index有签到信息对象存在就将适配器底色涂红，并显示学生姓名。

单击有学生坐的座位，进入查看学生详细资料/打分活动。

长按有学生坐的座位，跳出溢出菜单栏，里面有快速加分，快速扣分，显示详细信息三个按键。点击快速加分，访问服务器，修改表all\_check\_info中对应的课程和学生的属性score为5分（总分为5分，默认为3分）；点击快速扣分，访问服务器，修改表all\_check\_info中对应的课程和学生的属性score为1分（总分为5分，默认为3分）；点击显示详细信息，进入查看学生详细资料/打分活动。

如下图所示：

**图10：查看签到座位图活动界面**

3.3.2.5查看学生详细资料/打分活动

工具栏利用 toolbar组件，toolbar的title显示传来的学生信息对象的学号和姓名。

访问服务器，查询表student中对应的学生的属性student\_sex，student\_college和student\_class，利用textview组件显示；查询表all\_check\_info中对应的课程和学生信息中的属性score，利用ratingbar组件显示分数，教师点选五角星即实时更新数据库中的分数。

如下图所示：

3.3.2.6课程历史信息活动（教师）

利用textview组件显示这个班级应到人数，出勤人数，迟到人数和开放签到时间。班级应到人数是通过上一个活动中传递过来的课程信息得到，出勤和迟到都是通过访问服务器，查询表all\_check\_info中ischeck属性分别为“1、5”，“2、3、4”的信息数目得到。开放签到时间是通过查询表all\_check\_info中老师在这门课程的这节课上插入的签到信息中的check\_time即是开签时间而得到。

点击“查看学生列表”按钮，进入查看学生列表活动，同上述签到事件中的学生列表活动。

点击“查看签到座位图”按钮，进入查看签到座位图活动，同上述签到事件中的查看签到座位图活动。

如下图所示：

**图11：查看学生详细资料/打分活动 图12：课程历史信息活动（教师）界面**

3.3.3学生用户组活动

3.3.3.1选择需要签到的课程界面（fragment）

登录活动会向这个活动传一个字符串参数“user”，若它是“student”即为教师用户组则显示以下界面。

向服务器发起请求，查询当前教师用户执教的课程，利用listview组件按照“@课程名@课程编号@教室位置@教师姓名”的条目顺序显示。

单击某一门课程，进入那门课程的签到情况页面。

工具栏利用toolbar组件，上有刷新键和溢出菜单栏，内有退出登录和添加课程按键（但添加课程并不对教师开放）。

点击刷新建，向服务器发起请求，查询当前教师用户执教的课程，利用listview组件按照“@课程名@课程编号@教室位置@教师姓名”的条目顺序显示。

点击退出登录按键，进入登录活动，但会传输一个字符串参数“exit”，进入登陆活动后接收到字符串参数后会把已经填写过的表单信息清除掉。

点击添加课程按键，进入添加课程活动。

界面如下图所示：

**图13：选择需要签到的课程活界面**

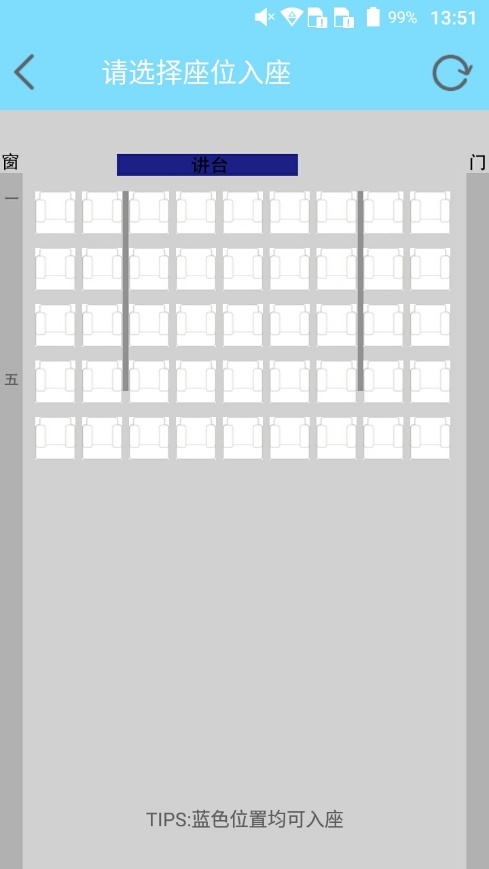
3.3.3.2查看教室座位图活动

工具栏利用toolbar组件，toolbar的title即时显示已签多少学生：访问服务器，查询all\_check\_info中属性subject\_id和subject\_th与这节课对应的信息条数即是这门课实到人数。工具栏上还有返回和刷新键。

工具栏下面的界面利用gridview组件显示了一个座位图。显示之前先访问服务器，查询表subject中这节课程对应的属性class\_type，根据class\_type显示小、中、大、特大尺寸的教室对应的座位图；查询表all\_check\_info中属性subject\_id和subject\_th与这节课对应的签到信息，使用json转换，恢复ArrayList数据结构，这个ArrayList存放着这个班的已签学生；创建一个大小为教室座位总数的布尔数组，以学生的座位号seat\_index为下标，若有人坐了就设置table[seat\_index]为true;再创建一个大小为教室座位总数的签到信息类数组，将座位图先抽象为一个数组，哪里坐了学生就下标对应的元素是那个学生的签到信息。这样一来，在构建座位图的循环中就不会有访问服务器请求，极大降低了时间成本。在显示座位图的时候若对应seat\_index有签到信息对象存在就将适配器底色涂红，并显示学生姓名。

单击有学生坐的座位，弹出警告框，告知学生这里有人了。

单击没有学生坐的座位，弹出警告框，询问学生是否在此入座，完成签到。若这是一个在教室后两排的座位，就会弹出警告框，告知学生不能在此入座。

界面如下图所示：

**图14：查看教室座位图活动界面**

3.3.3.3课程历史信息活动（学生）

利用textview组件显示学生在这节课上的签到状态（出勤/迟到/教师代签），入座座位号和签到时间。这3个信息都是通过访问服务器，查询表all\_check\_info中学生和课程对应的信息中的属性ischeck,seat\_index和check\_time。

工具栏利用toolbar组件，toolbar的title即时显示“第‘X’节XX课程”。

界面如下图所示：

**图15：课程历史信息活动（学生）界面**

3.3.4公共用户组活动

3.3.4.1启动动画活动

参考博客，详见http://blog.csdn.net/sinat\_28676875/article/details/68935611。

界面如下图所示：

3.3.4.2用户登录活动

利用editText组件输入id和password；用Checkbox组件勾选是否需要自动登陆，记住密码；用radiobutton组件点选用户组。

点击“登陆”按钮，向服务器发起登陆请求，若数据库中存在相关条目即登陆成功，进入主页面，同时提供错误回馈功能。（没有点选用户组，没有填写用户名或密码）进入主页面时向主页面传递参数：用户组字符串（用来判断登陆进来的是哪个用户，显示不同的界面）和对应的用户信息对象（储存最基本的id信息）。

点击“忘记密码”文字，进入忘记密码页面。

界面如下图所示：

**图16：启动动画活动界面 图17：用户登录活动界面**

3.3.4.3修改密码活动

读入用户组信息。利用edittext组件输入老密码，新密码和确认新密码，点击“确认”按钮，向服务器发起请求，如果用这个教师/学生的id和输入的旧密码能够完成登陆操作，那么就判断输入的新密码和确认新密码是否相等，若相等再判断新密码和旧密码是否相等，若不相等则将服务器数据库中表student/teacher中属性student\_password/teacher\_password替换成输入的新密码，同时提供各种错误提示如新密码和确认新密码不相等、新密码和旧密码相等。

界面如下图所示：

**图18：修改密码活动界面**

3.3.4.4修改个人资料活动

读入用户组信息，访问服务器，利用该用户的id查询表student/teacher中查询所有的属性（名字，性别，学院，班级（只有学生有班级这个属性）） ，利用textview组件都显示出来。用户点击某一个属性就进入相关活动进行修改。（只有性别、学院、班级可以修改）。

在修改性别和修改学院活动中，利用spinner组件让用户进行选择，点击确认后，访问服务器，替换掉原来数据库中存储的信息。

在修改班级活动中，利用edittext组件让用户自行输入，点击确认后，，访问服务器，替换掉原来数据库中存储的信息。

界面如下图所示：

**图19：修改个人资料活动界面**

3.3.4.5 App设置活动

利用switchbar组件控制相关设置功能的开启/关闭状态。利用properties配置文件存储相关设置信息。若系统监听到switchbar的状态发生了改变则进行相关值在配置文件中的修改。

界面如下图所示：

**图20：设置活动界面**

3.3.4.6添加课程活动

读入用户组信息，若是教师用户则弹出错误信息返回前一活动。（我们并没有提供教师添加课程的功能）

利用editText组件输入课程编号。

点击“添加”按钮，向服务器发起请求，查询表subject中是否存在用户输入的课程编号，若存在，则向表student\_xxxxxxx中插入这条课程编号，同时提供错误回馈功能。（没有填写课程编号）

界面如下图所示：

**图21：添加课程活动界面**

3.3.4.7忘记密码活动

利用edittext组件输入id和name，利用radiobutton选择用户组。单击“确定”按钮，向服务器发出请求，在对应的用户组信息表（student/teacher）表中查找属性为student\_id./teacher\_id和student\_name/teacher\_name对应相等的信息，若存在则利用textview输出用户的密码student\_password/teacher\_password，若没有找到则跳出错误信息。

界面如下图所示：

**图22：忘记密码活动界面**

3.3.4.8选择需要查看历史信息的课程界面（fragment）

登录活动会向这个活动传一个字符串参数“user”，若它是“teacher”即为教师用户组则显示以下界面。

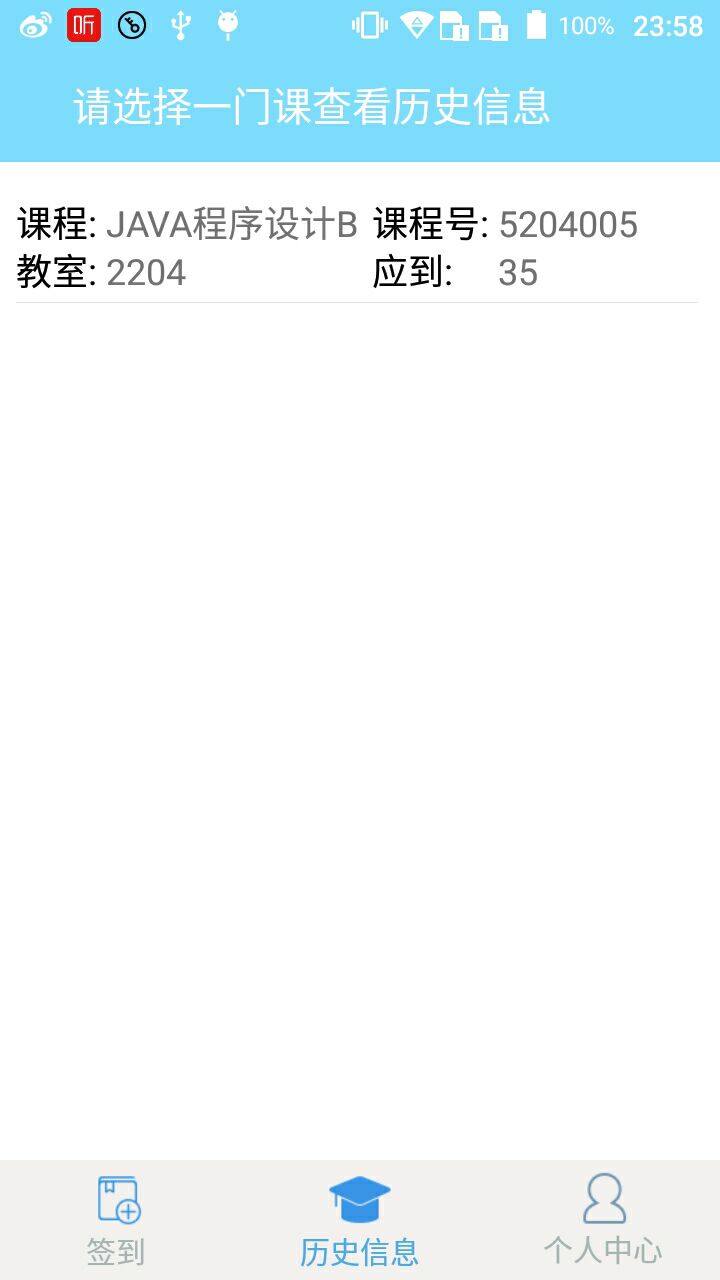
向服务器发起请求，查询当前教师用户执教的课程，利用listview组件按照“@课程名@课程编号@教室位置@学生人数”的条目顺序显示。

若user参数是“student”即为学生用户组则显示以下界面。

向服务器发起请求，查询当前学生用户参加的课程，利用listview组件按照“@课程名@课程编号@教室位置@教师姓名”的条目顺序显示。

单击某一门课程，进入那门课程的选择需要查看历史信息的课程中某堂课活动。

界面如下图所示：

**图23：选择需要查看历史信息的课程界面**

3.3.4.9选择需要查看历史信息的课程中某堂课活动

向服务器发起请求，查询当前教师/学生用户的课程，利用listview组件按照@日期@节数”的条目顺序显示。

单击某一“时间-节数”条目，进入课程历史信息活动。

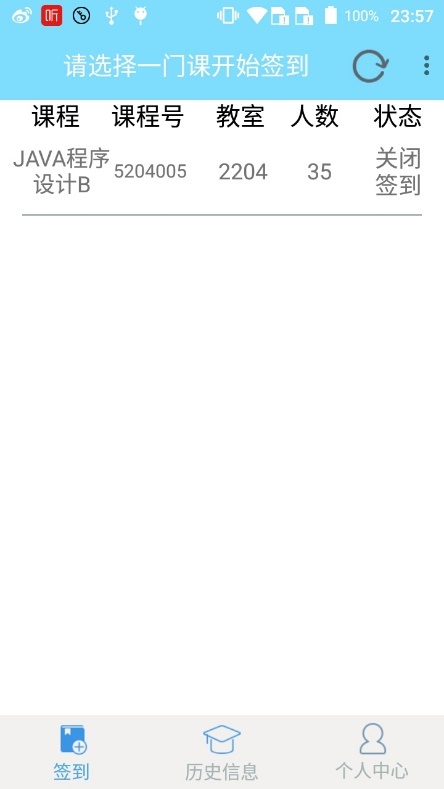
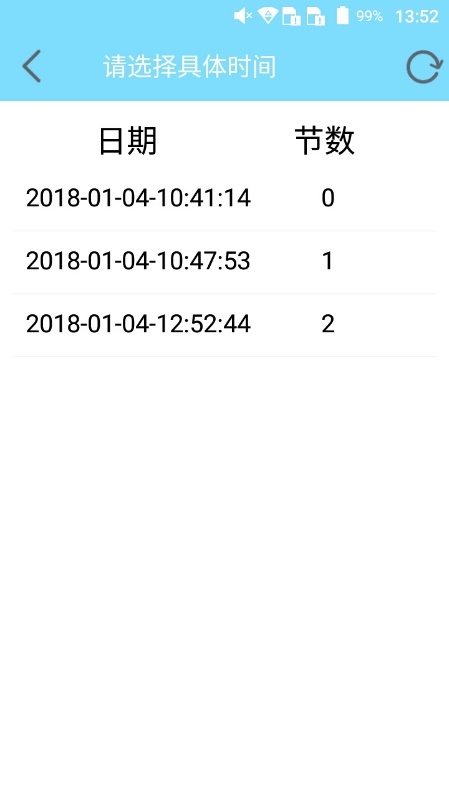
工具栏利用 toolbar组件，上有刷新键。

单击刷新键，向服务器发起请求，查询当前教师/学生用户的课程，利用listview组件按照@日期@节数”的条目顺序显示。

界面如下图所示：

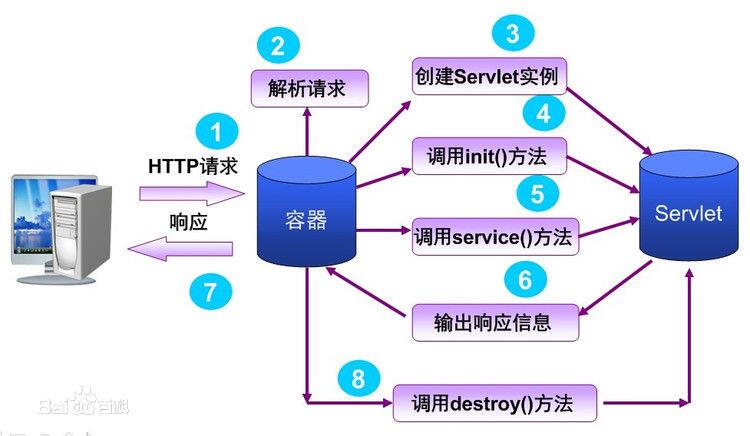
3.3.4.10App主导航活动

界面如下图所示：

**图24：选择需要查看历史信息的课程中某堂课活动******

**图25：主导航活动界面（默认为选择需要签到的课程界面）**

3.4服务端结构设计与实现

服务端利用servlet完成。由于是相关知识的初学者，所以编写servlet代码时保证每个servlet.java代码只完成一个访问数据库操作。这样一来导致服务端项目中servlet文件非常多，代码略有冗余，但简化了开发难度：在客户端中有一个数据请求就在客户端中写一个联网线程和一个使用线程的类，在服务端中写一个实现这个操作的servlet文件，有时候还需要在客户端中写相关json转化代码。所有的通信过程都是如此，大同小异，殊途同归。核心技术就是servlet和JDBC，其他逻辑都是普通的Java代码了。结构如下图所示。

**图26：Servlet体系结构图**

3.4.1数据库的增删改查

3.4.1.1增添信息 以InsertCheckInfo.java为例

InsertCheckInfo.java是学生用户调用的。当学生在查看教室座位图活动中选好了位置要入座时调用这个servlet，插入相关签到信息，完成签到。

学生用户提供“@课程编号@课程节数@学生学号@入座座位号@教师开始签到时间”信息，在doPost中以request接收。

利用Java的date类、SimpleDateFormat类，获取服务器本地时间与教师开始签到时间相减，得到的结果若小于10分钟则学生签到信息条目中的ischeck属性为1，小于40分钟大于10分钟则为2，小于90分钟大于40分钟则为3，大于90分钟则为4。（以上时间均可以方便修改）

利用JDBC执行以下SQL语句："insert into all\_check\_info"+ "(subject\_id,subject\_th,student\_id,seat\_index,ischeck,check\_time)" + "values" + "('" + subject\_id+ "'," + subject\_th\_num + ",'" + student\_id + "','" + seat\_index\_num + "'," + ischeck + ",'"+ check\_time + "')"，若出现异常则向客户端返回fail语句，否则返回success语句。在客户端中执行相关判断语句。

3.4.1.2删除信息 以DeleteCheckInfo\_Teacher.java为例

DeleteCheckInfo\_Teacher.java是教师用户调用的。当教师在开启签到过程中碰到了误操作后（在错误的节数开启了签到，而开启签到会插入一条存有开签时间的信息。由于主键存在且唯一，若不删除会影响真正到这节课时的签到。）要把节数调回来，这时候就会调用这个servlet。

教师用户提供“@课程编号@课程节数@教师工号”信息，在doPost中以request接收。

利用JDBC执行以下SQL语句："DELETE FROM `iclass`.`all\_check\_info` WHERE `subject\_id`='" + subject\_id+ "' and`subject\_th`='" + subject\_th\_num + "' and`student\_id`='" + teacher\_id + "'"，若出现异常则向客户端返回fail语句，否则返回success语句。在客户端中执行相关判断语句。

3.4.1.3修改信息 以UpdateSubjectTh.java为例

UpdateSubjectTh.java是教师用户调用的。教师要开始签到前，要确认当前课程已经上到第几节了。上完一节课后教师要自行让这个节数自增一节，这个时候要调用这个servlet。

教师提供“@课程编号”信息，在doPost中以request接收。

利用JDBC执行以下SQL语句："UPDATE subject SET subject\_th=subject\_th+1 " + "WHERE subject\_id ='" + subject\_id + "'"，若出现异常则向客户端返回fail语句，否则返回success语句。在客户端中执行相关判断语句。

3.4.1.4查询信息 以GetTeacherSubject.java为例

GetTeacherSubject.java是教师用户调用的。教师在选择需要签到的课程活动中的listview显示的课程信息来源就是这个servlet。

教师提供“@教师工号”信息，在doPost中以request接收。

利用JDBC执行以下SQL语句："select \* from subject where teacher\_id= '" + teacher\_id + "'"。

利用JSON，将得到的结果中的每个属性都放入JSONObject中，再将每个JSONObject放入JSONArray中。

利用URLEncoder编码UTF-8后，将得到的JSONArray传回客户端，类型为String。

在客户端中，利用URLDecoder解码得到中文字符，将JSONArray中每一个JSONObejct恢复数据结构，恢复为课程信息对象，再放入ArrayList中，就能直接在listview组件的显示代码中使用了。

4总结

【对作品的创意、开发实现过程进行总结，给出作品的改进方向。】以下内容为直接采访文稿：

4.1孙翊佳谈作品的创意

我想到做这个App的时候，首先是在当时上英语课的时候，老师就是通过自己在A4纸上画一个座位图，他自己用手画的。她让我们每个人把名字都写在上面，但用这种签到方式，我感觉虽然说实用性较强，老师能直观地看出这节课哪些学生来了，哪些学生没来，并能直接，及时地给学生进行打分，考评。关键我想每节课都通过手写的方式进行签到，效率其实是挺低的，因为每次一张纸转一圈，我们至少要花上一刻钟的时间。那其他老师点名的话，通过名单逐次进行点名，这个其实也是相当耗功夫。有的老师花5分钟点完，有的老师要花上15分钟进行才能点完。

所以我想到就是，把老师手动的添加学生到他自己纸张上的过程，搬到手机上面来。这样既方便了学生也方便老师——老师能够看到哪些学生迟到，哪些学生是正常来上课的，大大简化每节课的签到工作。

4.2蔡佳昊谈开发实现过程

4.2.1万事开头难。

首先，我们3个人的本科专业本行都不是做软件开发，特别是移动应用开发。我是计算机科学与技术CS专业的，大一第二学期立项的时候我连Java都还没有接触过；同事孙翊佳和我一届，是信息与计算科学专业的，可以说是纯正的“数学家”了，和埋头编码都沾不上边，但整个项目的想法都是他想出来的。还有同事马耀是大一届的学长，负责数据库的他是是空间信息与数字技术专业的。

还好我和马耀都长时间准备过各种程序设计比赛、考试，编码能力有一定保障。这让我们后面的编码过程能大步流星地进行。

4.2.2碰到了几个比较大的困难。

第一个就是数据库选错了，一开始我们选择了轻量级的SQLite，这个数据库没办法和服务器交互，或者说我们还不会和服务器交互。后来我们选择了在服务端用MySQL，本地没有数据库。

第二个问题就是通信交互的问题，怎么实现客户端和服务端的通信，以实现我们的业务逻辑。这个问题卡了很久很久很久很久，9月中到11月初那么久的时间都在干耗，其中有一些泄气，没信心了的原因。而且在我们一开始计划的时候，完全没有考虑到这个问题会有那么的困难。对我们来说，服务端JavaEE的内容过于陌生，如果不是做找个App可能我本科毕业了学校课程都不会教怎么搭建服务器，怎么写服务端程序。还好责任心使然，纯靠自己琢磨，根据网上的一些样例，最后终于解决了这个技术难点，还是非常骄傲的。

还有个问题，就是Android入门书籍上学到的东西和这种企业级App开发的要求极不匹配。我看的那本入门书籍连navigation bar、toolbar的使用都没有提及，导致我们最初在UI上绕了很多弯路。我们几个男生也缺少UI界面设计的经验，做出来的App至始至终在UI上和手机上用的大公司上线的App有天差地别。

4.2.3App核心业务逻辑

现在回过头来看其实还是挺简单的，就是在服务器数据库上不断地进行读写操作实现的。老师先写（开放签到），学生读了（读取开始签到时间，签到状态开放与否）再写（插入签到信息），老师再不停地读（读取已签学生，显示表、图）。

App的特色功能这种座位图，当初想到要做这个的时候感觉特别难实现，因为脑子里一直对“图”有芥蒂。后来，后来真的做了感觉也就那么回事儿。看着网上九宫格的例子，再稍微结合一下自己的需求就把班级座位图捯饬出来了。

这期间还结合了学数据结构与算法的时候的思想。比如上文提及的实现座位图逻辑，有一个问题就是：如何获知每个座位上到底有没有学生坐呢？已知的条件是学生的seat\_index即座位号。我想到了平时C++编程中常用的方法：“bool table[MAX](false);（C++11）”创建一个大小为教室座位总数的布尔数组，以学生的座位号seat\_index为下标，若有人坐了就设置table[seat\_index]为true;再创建一个大小为教室座位总数的签到信息类数组，将座位图先抽象为一个数组，哪里坐了学生就下标对应的元素是那个学生的签到信息。这样一来，在构建座位图的循环中就不会有访问服务器请求，极大降低了时间成本。

4.3马耀谈团队合作心得体会

现在手机已经非常普及，基本上随便找一个人都有自己的手机，所以就想利用好这一点，为我们的教学活动服务。正好在我在寻找队友的了解到现在团队里其他成员也在做一个无纸化考勤的Android App,所以就加到他们团队中了，正好我们的专业知识技能也都能够互补，有人熟悉UI界面，有人熟悉Android编程，而我作为团队里学过数据库原理的成员，自然数据库那一部分就由我来组要负责完成了。我们的想法是做一个围绕考勤展开的软件，主要是因为现在这一方面的软件确实不多，这个领域还有待完善。现在上课基本上都是采用结合多媒体的形式，但是考勤这一方面一直以来都是纸质考勤表，考勤非常浪费时间，而且现在是出勤信息不仅仅需要任课老师需要知道，还需要有各个班级内有同学负责统计出勤信息报给辅导员，这样有时候就会造成重复统计的问题。我们就像将所有信息集成到一个平台上，无论是任课教师还是辅导员等等，都可以通过这个平台来查看出勤情况。而且签到由请求由学生发起，这样就可以节省任课老师等考勤造成的时间浪费。

团队合作嘛，不是一个人在做事，相互写作业就非常重要了，一开始的时候，我们基本上是所有人都在做同一件事情，但是后来发现不同人做的东西会有出入，毕竟每个人的想法都是不一样的，本身实现同一个程序这个过程不同人写出来的也都是五花八门，这就影响到了我们的效率。之后呢，就干脆每人负责一部分。

做这个项目用到的知识并不都会，对于我们现在也很难完全掌握，素以分工就很重要了。即使这样我们还是要去学习很多新内容，当然，也会碰到很多麻烦。就拿我负责的部分来说，一开始我也有参与到通信那一部分，也在网上搜索，跟着动手做一做，这部分就是一个之前未接触到的领域。道到解决问题这个难点占用了我们很多的时间内，这也促成了我们之后的分工。

参与这种项目，我们也都是每人都贡献一点想法，然后在慢慢完善改进。我认为想法是非常重要的，至于技能也就排在其后。实现上我们多花点时间查查资料就也能够搞定，但是呢，没有想法的话一个项目就停滞了。

我是收获很多的，参与这个项目让我对Android、Servlet等等的内容有了了解，并且呢,我现在也能够运用这些东西了，对我来说这就是进步，很享受这个过程。

三人谈作品的改进方向

直接与教务处进行数据交互，彻底实现全自动

直到现在仍困扰着我们的是数据库的数据来源。我们希望能与校方教务处的学生、课程数据实现互通，减少管理员、老师或者学生的操作。

实现企业级App要求：更优秀的用户体验；完善UI设计；全尺寸设备支持；全平台支持；

由于知识和精力有限，我们的App目前只能支持5.0英寸的Android手机正常显示。

进行教学楼三维建模，教室座位与二维码一一对应，学生一秒扫码签到，彻底实现签到和判断的自动化

整合大学校园各种需求，实现校园生态一体化App

5附录

5.1参考资料

5.1.1书籍

Android Studio开发实战 从零基础到App上线 ISBN：978-7-302-47006-9

Android APP开发入门：使用Android Studio环境 ISBN：978-7-111-53958-2

SQL入门经典（第五版） ISBN：978-7-115-26407-7

JSON必知必会 ISBN：978-7-115-42207-1

ExtJS+Android+SSH整合开发Web与移动SNS ISBN：978-7-121-11167-9

ACM/ICPC程序设计与分析（C++实现） ISBN：978-7-302-22373-3

5.1.2博客

1、写一个简单炫酷的app程序的打开动画

http://blog.csdn.net/sinat\_28676875/article/details/68935611

2、SQLite数据库内容显示到ListView http://www.cnblogs.com/soada/p/5705867.html

3、android+eclipse+mysql+servlet(Android与mysql建立链接)

http://www.cnblogs.com/zhu520/p/7724524.html

4、Android——Tomcat+MySQL+Servlet，实现将Client传入的数据写入MySQL

https://www.2cto.com/kf/201604/497541.html

5、使用MySQL和Servlet编写Android接口样例<http://blog.csdn.net/u010368726/article/details/50686761>

6、AndroidTomcat+MySQL+Servlet 实现将Client传入的数据写入MySQL <http://www.it165.net/admin/html/201604/7323.html>

7、Tomcat数据库连接池配置<http://blog.csdn.net/qq_24421591/article/details/51055390>

8、MySQL数据库安装与配置详解<http://www.cnblogs.com/sshoub/p/4321640.html>

9、JSON在Java中的使用

http://www.cnblogs.com/xue6666/p/5917185.html

10、Android Studio 统计项目的代码总行数<http://blog.csdn.net/u012145166/article/details/51057102>

11、使用 URLDecoder 和 URLEncoder 对中文字符进行编码和解码http://blog.csdn.net/justloveyou\_/article/details/57156039

12、在android程序中使用配置文件properties

http://blog.csdn.net/jdfkldjlkjdl/article/details/46831219

13、安卓情景模式开发（二）－控制静音/音量/振动

https://www.cnblogs.com/wii/archive/2012/03/20/2408257.html

14、In Android 7 (API level 24) my app is not allowed to mute phone (set ringer mode to silent)

https://stackoverflow.com/questions/39151453/in-android-7-api-level-24-my-app-is-not-allowed-to-mute-phone-set-ringer-mode

15、Android中生成excel

http://blog.csdn.net/u014486880/article/details/50605659

16、获取Android设备的唯一标识符

http://blog.csdn.net/sunsteam/article/details/73189268

17、Android保证首次获取到的location对象不为空的解决方案

http://blog.csdn.net/zhangyu\_kenshin/article/details/50651409

5.1.3专业术语

5.1.3.1活动Activity

一个Activity是一个应用程序组件，提供一个屏幕，用户可以用来交互为了完成某项任务，例如拨号、拍照、发送email、看地图。每一个activity被给予一个窗口，在上面可以绘制用户接口。窗口通常充满屏幕，但也可以小于屏幕而浮于其它窗口之上。

5.1.3.2布局Layout和其xml文件

布局决定了Activity所展现的样子，他决定了布局的结构和控制着展现给用户所有的元素。采用xml文件布局的好处就是可以将需要显示元素从控制层的代码中分离出来，描述UI的部分和应用的代码是分离的，这让修改这些布局文件时不需要考虑去修改应用的代码并且重新编译。

5.1.3.3 JDBC

JDBC（Java DataBase Connectivity,java数据库连接）是一种用于执行SQL语句的Java API，可以为多种关系数据库提供统一访问，它由一组用Java语言编写的类和接口组成。JDBC提供了一种基准，据此可以构建更高级的工具和接口，使数据库开发人员能够编写数据库应用程序。

5.1.3.4本App中用到的Android控件简介

##### RelativeLayout相对布局

RelativeLayout下级视图的位置是相对位置，得有具体的参照物才能确定最终位置。

##### LinerLayout线性布局

LinerLayout下面的子视图就像用一根线穿了起来，所以LinerLayout内部视图的排列是有顺序的，要么从上到下依次垂直排列，要么从左到右依次水平排列。

##### Edittext文本编辑框

用户可在此输入文本等信息。

##### Textview文本视图

TextView是最基础的文本显示控件。

##### ImageView图像视图

ImageView是图像显示控件。

##### Button按钮

Button派生自TextView，二者在UI上的区别主要是Button控件有个按钮外观，提示用户点击这里。

##### ListView列表视图

ListView是常见的座位图视图，允许在页面上分行展示相似的数据界面，如新闻列表、商品列表、书籍列表等，方便用户逐行浏览与操作。

##### GridView网格视图

GridView是常见的适配器视图，用于分行分列显示表格信息，比ListView更适合展示商品清单。

##### Spinner下拉框

Spinner用于从一串列表中选择某项，功能类似于单选按钮的组合。

##### RadioButton单选按钮

RadioButton单选按钮要在一组按钮中选择其中一项。

##### Switch开关按钮

Switch是一个高级版本的CheckBox，在选中与取消选中时克展现的界面元素比CheckBox丰富。

##### Checkbox复选框

CheckBox是抽象类CompoundButton类一个最简单的实现，点击复选框勾选，再次点击取消勾选。

##### Toolbar工具栏

在Android5.0推出了Toolbar，意在取代ActionBar。主流App除了底部有一排标签栏外，通常顶部还有一排导航栏。这个导航栏在Android5.0之前以ActionBar控件的形式出现，但ActionBar存在不灵活、难以拓展的毛病。

##### Fragment碎片

Fragment是个特别的存在，有点像报纸上的专栏，看起来只占据页面的一小块，但是这一小块有自己的生命周期，可以自行其事，仿佛独立王国；并且这一小块的特性无论在哪个页面给一个位置就行，添加后不影响宿主页面的其他区域，去除后也不影响宿主页面的其他区域。

##### Navigation Bar导航栏

提供一种机制来显示一个导航栏下方的状态栏。

5.2源代码清单

详见源代码文件夹中的文件，客户端项目文件夹名为iClass，服务端项目文件夹名为iClass\_Sever。