<到云>移动端产品需求说明书

小组成员：连思璜、李胜亮、李豪、王耀鹏、刘灵清

版本修改页

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修订人** | **修订日期** | **修订描述** |
| V01 | 全体 | 2020.02.25 | 制定一个粗略的移动端需求分析 |
| V02 | 连思璜 | 2020/3/18 | 增加教师查看班课成员功能 |
|  |  |  |  |
|  |  |  |  |

目录

目录

[1. 引言 1](#_Toc34426559)

[1.1. 目的 1](#_Toc34426560)

[1.2. 背景 1](#_Toc34426561)

[1.3. 定义 1](#_Toc34426562)

[2. 项目概述 2](#_Toc34426563)

[2.1. 产品结构图 2](#_Toc34426564)

[2.2. 产品信息结构图 3](#_Toc34426565)

[2.3. 用户 4](#_Toc34426566)

[2.4. 业务流程 4](#_Toc34426567)

[2.5. 全局说明 5](#_Toc34426568)

[3. 功能详细需求 5](#_Toc34426569)

[3.1. 班课 5](#_Toc34426570)

[3.1.1. 加入班课 5](#_Toc34426571)

[3.1.2. 创建班课 6](#_Toc34426572)

[3.1.3. 查看班课 7](#_Toc34426573)

[3.1.4. 搜索 9](#_Toc34426574)

[3.1.5. 签到 9](#_Toc34426575)

[3.2. 我的 11](#_Toc34426576)

[3.2.1. 我的主页 11](#_Toc34426577)

[3.2.2. 用户信息 11](#_Toc34426578)

[3.2.3. 帮助 12](#_Toc34426579)

[3.2.4. 关联手机/邮箱 13](#_Toc34426580)

[3.2.5. 修改密码 14](#_Toc34426581)

[4. 非功能需求 15](#_Toc34426582)

[4.1. 观感 15](#_Toc34426583)

[4.2. 易用性 15](#_Toc34426584)

[4.3. 安全性 16](#_Toc34426585)

[4.4. 可维护性 16](#_Toc34426586)

# 引言

## 目的

本文档是“到云”系统的需求说明文档，主要作为确认需求以及系统分析设计的依据。文档详尽说明了这一软件产品的需求和规格，这些规格说明是进行设计的基础，也是编写测试用例和进行系统测试的主要依据。同时，该文档也是用户确定软件功能需求的主要依据。

## 背景

在高校里,课堂考勤在教学管理中至关重要。课堂考勤情况通常与学生成绩息息相关。高校中常使用的考勤方式,主要是教师点名和学生签到,这样的方式既浪费课堂时间又效率低，还容易出现各种问题。并且考勤结果需要进行人工录入统计,麻烦且费时费力,很容易出现误差。因此，将“互联网+考勤”成为了大势所趋。

随着科技更新换代,各种签到方案竞相涌出。例如打卡考勤和指纹考勤,此两种考勤方式在某种程度起到了作用,但是由于教室众多,且学生多时造成拥挤,不能真正普及;此外,基于人脸识别的课堂签到系统,需要进行大量的图片识别、处理,而大量图像传输流量并不适用于数量巨大的实时签到。所以设计实现一个方便快捷、适合高校使用的签到系统亟待解决。

因此， 本次待开发的软件为课堂签到系统。但这个系统不仅仅只能用来签到，还能记录学生们的课堂表现，帮助老师更加直观高效地管理课堂。

## 定义

**前端**

Vue框架：是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与[现代化的工具链](https://cn.vuejs.org/v2/guide/single-file-components.html)以及各种[支持类库](https://github.com/vuejs/awesome-vue#libraries--plugins)结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

Element-UI：基于Vue2.0开发的一个组件库,提供了丰富的PC端组件。

**后端**

Node.js:是一个基于 Chrome V8 引擎的 [JavaScript](https://baike.baidu.com/item/JavaScript/321142) 运行环境。 Node.js 使用了一个事件驱动、非阻塞式 I/O 的模型。Node 是一个让 JavaScript 运行在[服务端](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E7%AB%AF/6492316)的开发平台，它让 JavaScript 成为与[PHP](https://baike.baidu.com/item/PHP/9337)、[Python](https://baike.baidu.com/item/Python/407313)、[Perl](https://baike.baidu.com/item/Perl/851577)、[Ruby](https://baike.baidu.com/item/Ruby/11419) 等服务端语言平起平坐的[脚本语言](https://baike.baidu.com/item/%E8%84%9A%E6%9C%AC%E8%AF%AD%E8%A8%80/1379708)。发布于2009年5月，由Ryan Dahl开发，实质是对Chrome V8引擎进行了封装。Node对一些特殊用例进行优化，提供替代的[API](https://baike.baidu.com/item/API/10154)，使得V8在非浏览器环境下运行得更好。V8引擎执行Javascript的速度非常快，性能非常好。Node是一个基于Chrome JavaScript运行时建立的平台， 用于方便地搭建响应速度快、易于扩展的网络应用。Node 使用[事件驱动](https://baike.baidu.com/item/%E4%BA%8B%E4%BB%B6%E9%A9%B1%E5%8A%A8/9597519)， 非阻塞[I/O](https://baike.baidu.com/item/I/O/84718) 模型而得以轻量和高效，非常适合在分布式设备上运行数据密集型的实时应用。

**数据库**

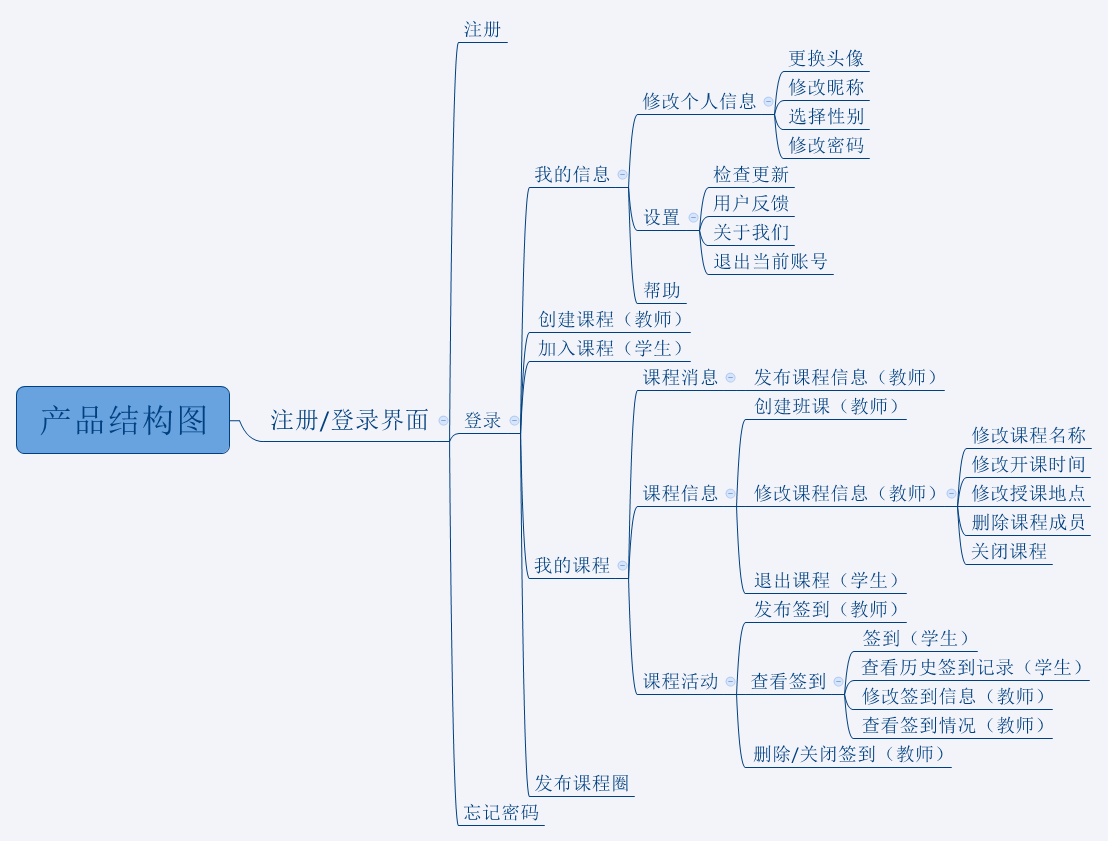
MySQL：是最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的 RDBMS(Relational Database Management System：关系数据库管理系统)应用软件之一。关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL所使用的 SQL 语言是用于访问[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728)的最常用标准化语言。由于其体积小、速度快、总体拥有成本低，尤其是[开放源码](https://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E6%BA%90%E7%A0%81/7176422)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。

# 项目概述

## 产品结构图

Table1.产品功能表

|  |  |  |
| --- | --- | --- |
| 频道 | 页面 | 说明 |
| 我的 | 我的主页 | 展示用户的昵称、积分等信息 |
| 帮助 | 软件使用教程 |
| 用户信息 | 用户的具体信息 |
| 设置 | 检查更新、关于我们等等 |
| 发现 | 课程圈 | 用户可以在这发布消息 |
| 班课 | 我加入的班课 | 学生加入 |
| 我创建的班课 | 老师已创建 |
| 搜索 | 搜索与自己有关的班课 |
| 创建班课 | 老师新创建 |

Fig 1.产品结构图

## 产品信息结构图

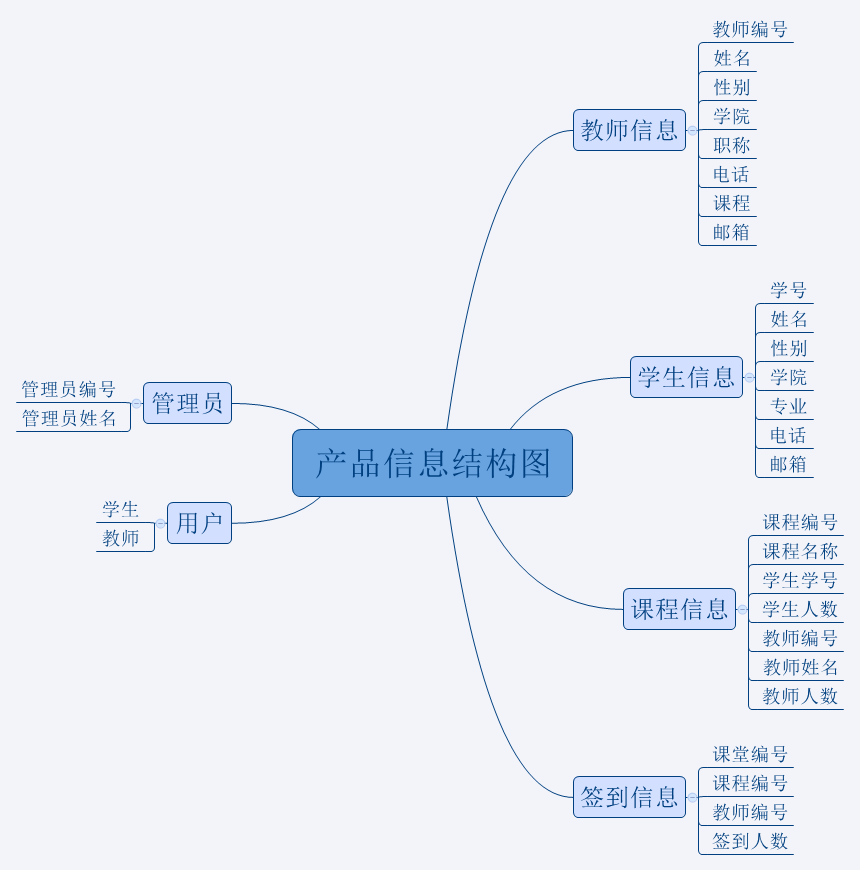


Fig 2.产品信息结构图

## 用户

Table 2.角色功能表

|  |  |  |
| --- | --- | --- |
| 角色 | 功能 | 特点 |
| 老师 | 创建班课、发布课程消息、修改课程、发布签到、查看学生信息、修改个人信息 |  |
| 学生 | 加入班课、查看消息、签到、修改个人信息 |  |



Fig 3.教师学生用例图

## 业务流程

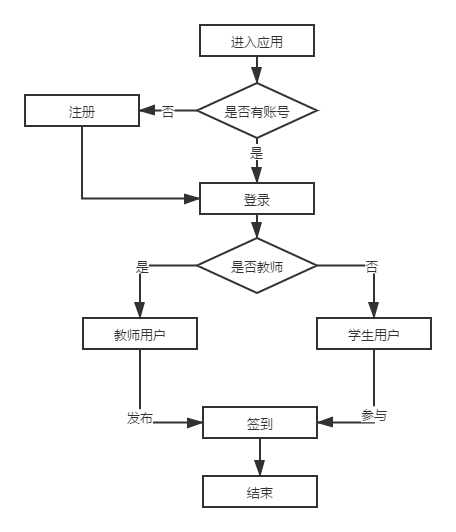


Fig 5.业务流程图

## 全局说明

本系统中，所有用户都需要登录后才能进入系统进行操作，因此如果没有账号，则会被要求注册。在登录页面有忘记密码选项，用户可以根据自己的预留信息修改密码。对于每个用户而言，都能修改自己的个人信息，例如修改昵称、更换头像等。

# 功能详细需求

## 班课

### 加入班课

用户场景：学生用户点击“班课”页面右上角的“+”，选择“使用班课号加入班课”，页面上会增加这个新加入的班课选项卡。

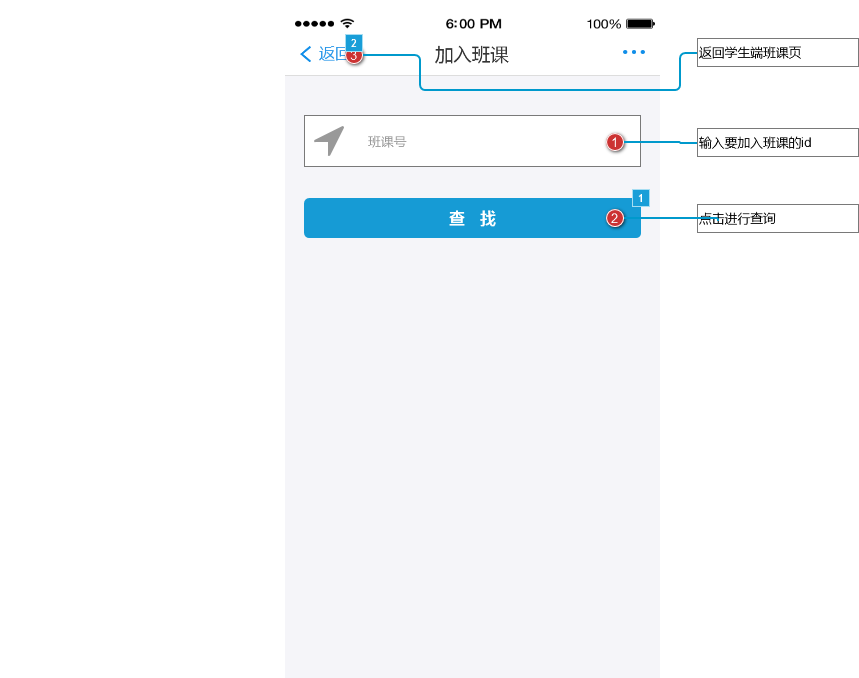
优先级：高

业务流程：无

输入/前置条件：教师需要先创建班课

页面逻辑

界面原型





交互：用户输入班课号，页面显示相应的课程信息，点击“加入班课”，完成加入。

字段：学生用户加入班课后，会自动将新增的学生信息加入到学生名单中

输出/后置条件：学生的“我加入的”会显示新增的这一门课程

### 创建班课

用户场景：教师用户点击“班课”页面右上角的“+”，选择“创建班课”，页面会跳转到创建新课程的页面。教师输入相关信息，点击“创建完成”。

优先级：高

业务流程：无

输入/前置条件：教师用户、管理员用户

页面逻辑

界面原型：



交互：教师用户点击“班课”页面右上角的“+”，选择“创建班课”，页面会跳转到创建新课程的页面。教师根据提示输入课程名称，上课时间等信息，最后点击“创建”，完成班课创建。页面提示“创建成功”。

字段：教师创建新的班课后，相关的课程编号、课程名称等信息会保存到相应的数据表中，新建的班课会关联创建者的教师id。

输出/后置条件：教师创建的课程会出现在“我创建的”中；如果是学生用户点击“创建班课”，会提示没有相关权限

### 查看成员

用户场景：教师用户点击“班课”页面的“我创建的”，点击班课标签页，跳转出班课成员的信息；再点击其中一个成员标签页，可以看到该成员的详细信息

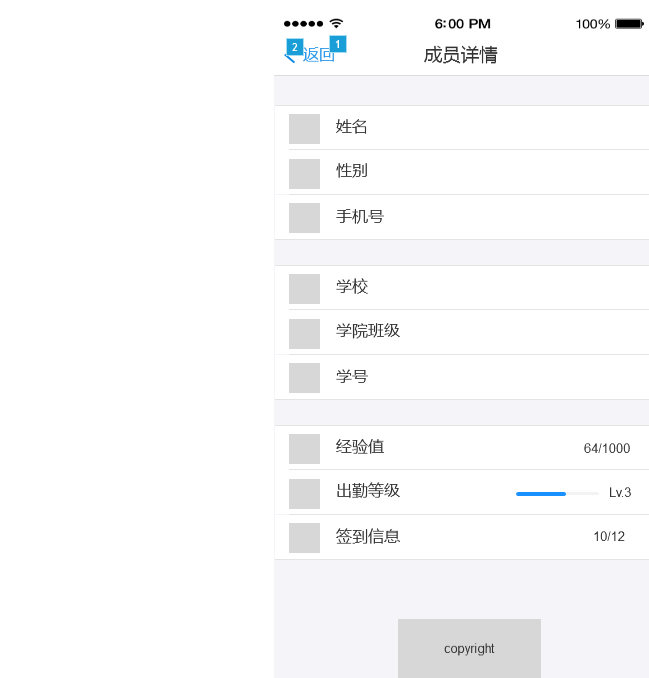
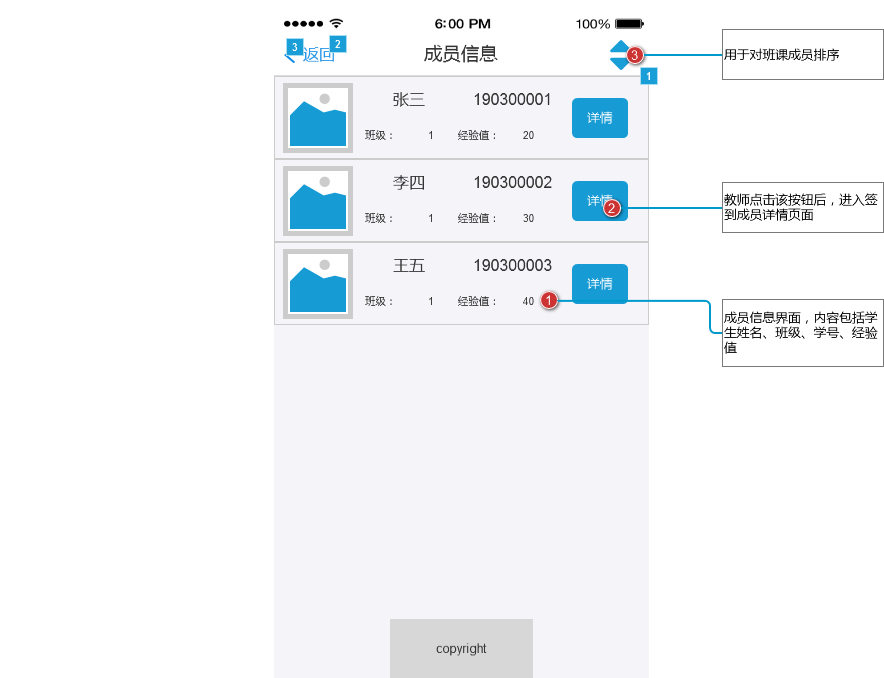
优先级：中

业务流程：无

输入/前置条件：班课需要提前创建；成员信息已经保存在后台

页面逻辑

界面原型：



交互：点击“详情”按钮后跳转到成员详情

字段：班课信息、成员信息从后台数据库获取

输出/后置条件：无

### 查看班课

用户场景：学生用户点击“班课”页面的“我加入的”，可以查看已加入的班课信息；教师用户点击“班课”页面的“我创建的”，可以查看已创建的班课信息。

优先级：高

业务流程：无

输入/前置条件：无

页面逻辑

界面原型：



交互：

字段：班课信息从后台数据库获取

输出/后置条件：无

### 搜索

用户场景：用户在搜索输入框中输入关键词，搜索包含该关键词的课程。如果有符合的结果，显示在页面上；如果没有，则显示没有找到

优先级：中

业务流程：无

输入/前置条件：如果想找到，对于教师而言，课程需要已创建；对于学生而言，需要加入课程，没有加入的课程是搜索不出来的

页面逻辑

界面原型：



交互：用户输入关键词/关键字，点击“搜索”，页面显示相应结果

字段：用户输入的内容用于比对，获取相应结果

输出/后置条件：显示相应结果，即使没有匹配的课程，也要有相应的提示

### 签到

用户场景：1、教师用户创建签到 2、学生用户点击进行签到

优先级：高

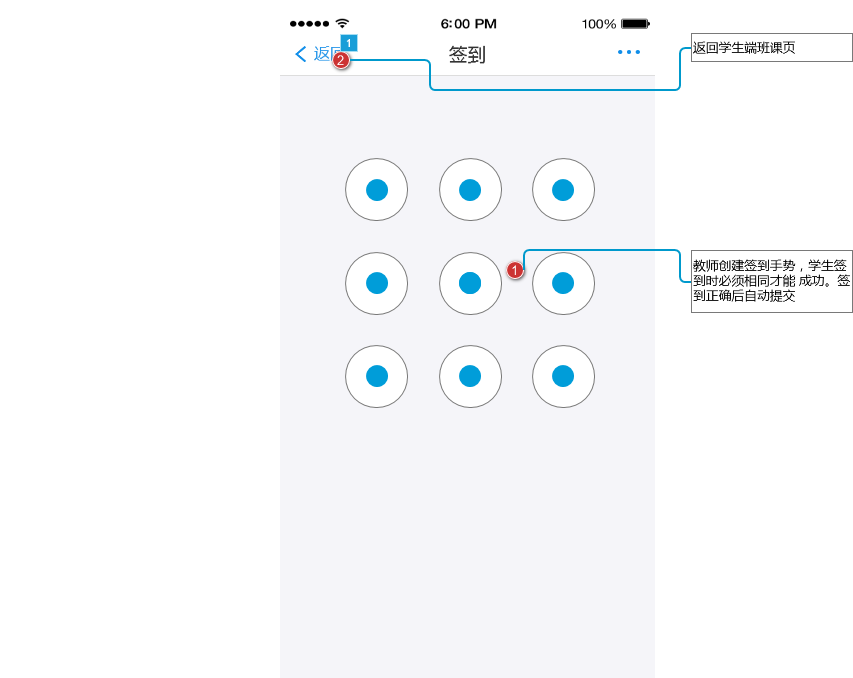
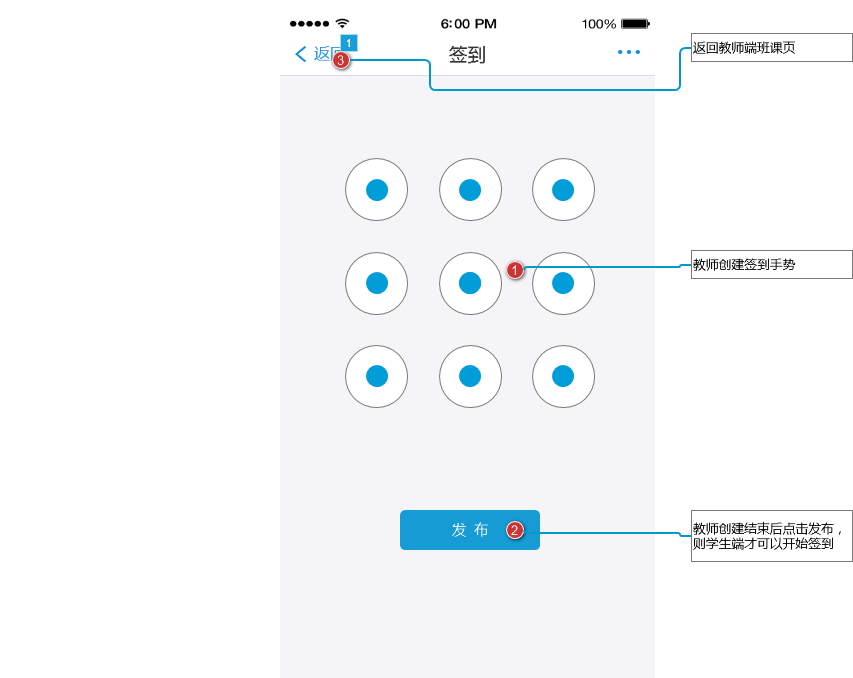
业务流程：无

输入/前置条件：1、创建签到的是教师用户

2、学生签到需要老师先创建签到，并且在签到的有效时间范围内

页面逻辑

界面原型：



交互：1、老师点击“开始签到”，页面显示“签到开始”，并开始倒计时。签到结束，教师可以看到学生的签到情况。

2、学生点击签到页面的“签到”按钮，签到成功会提示“签到成功”，并且这条签到记录会显示在签到页面中。

字段：学生签到成功后，这条签到记录会显示在相应的数据表中

输出/后置条件：无

版本：

## 我的

### 我的主页

用户场景：用户点击“我的”频道后显示的页面，会显示用户的头像、昵称，所在院校、

学院、专业等信息。帮助、设置等选项也包含在这个页面中。

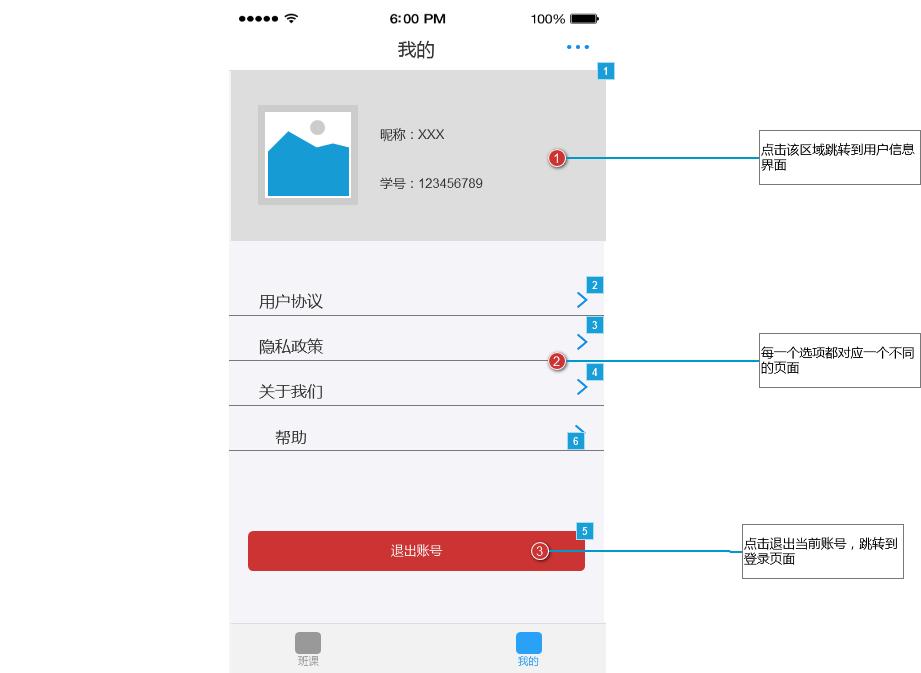
优先级：高

业务流程：无

输入/前置条件：无

页面逻辑

界面原型：



交互：无

字段：界面读取的信息均从数据库中获取

输出/后置条件：无

版本

### 用户信息

用户场景：用户点击“我的”主页中的昵称栏，跳转到该页面。用户可以修改个人信息，例如昵称、学号、用户身份等，点击“保存”完成修改。

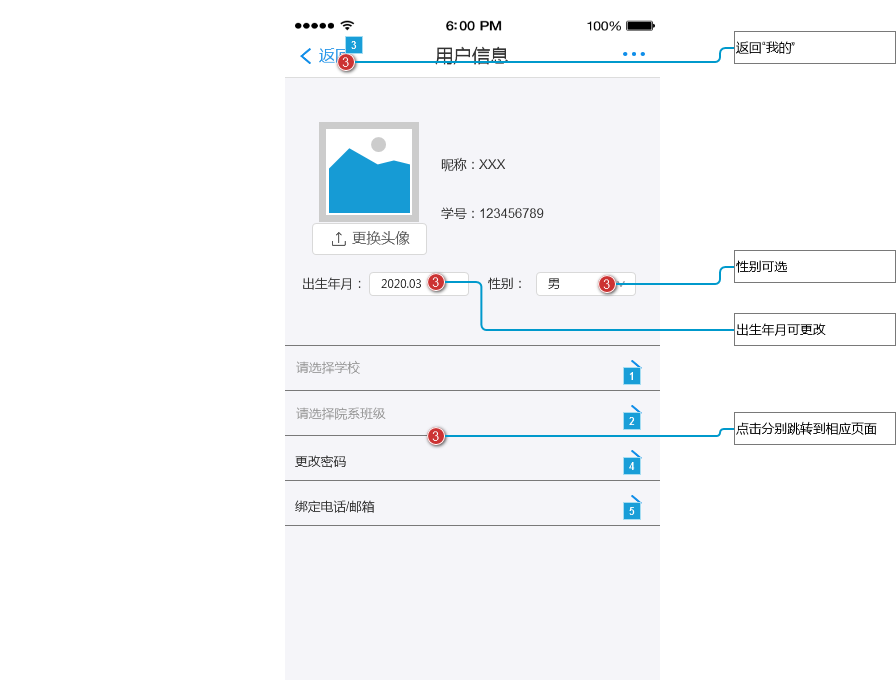
优先级：高

业务流程：无

输入/前置条件：数据库中有该用户的数据信息

页面逻辑

界面原型：



交互：用户点击相应的位置有相应的反应。例如，选择性别或选择身份，相应的选择后面有打钩，另外的则没有。修改个人信息后，点击“保存”，将数据库中的信息加以修改。

字段：显示的数据都从数据库中读取，修改后的数据保存到数据库中。

输出/后置条件：点击“保存”后，跳出提示“修改成功”。

版本

### 帮助

用户场景：用户在遇到困难时，点击 “帮助”，会跳转到帮助文档页面。

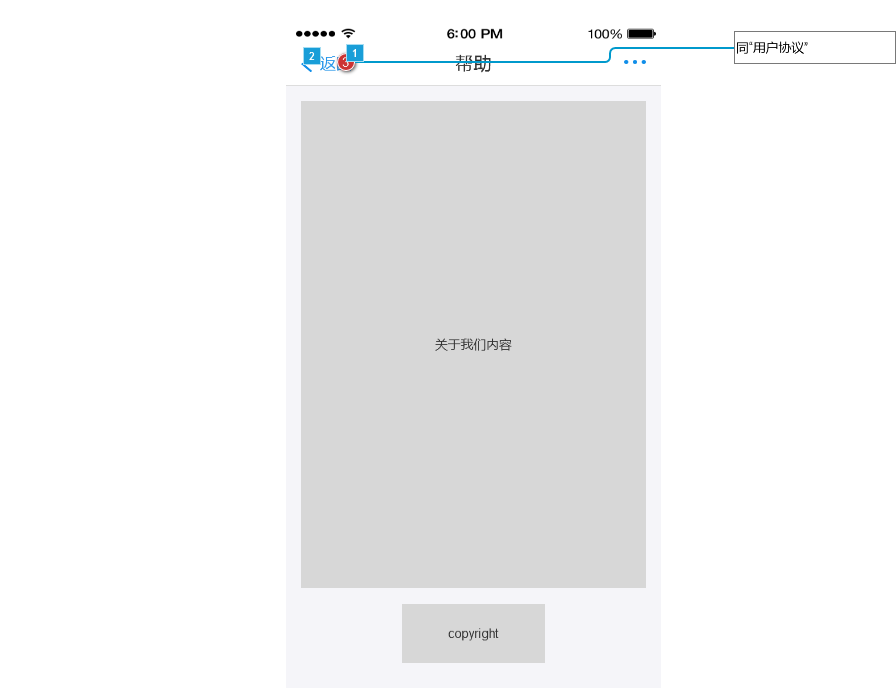
优先级：中

业务流程：无

输入/前置条件：该文档是自应用面世就存在的，是技术人员对于一些主要问题的提炼。

页面逻辑

界面原型：



交互：用户点击右上角的“帮助”，会跳转到帮助文档页面。

字段：帮助文档在后台数据中存在

输出/后置条件：无

版本

### 关联手机/邮箱

用户场景：该功能在设置中的账号安全里。点击“关联手机”/“关联邮箱”，如果没有关联过，显示的是“关联手机”/“关联邮箱”，否则显示“更改关联手机”/“更改关联邮箱”。然后进行身份认证，再输入手机号码/邮箱地址，获取验证码，完成关联。

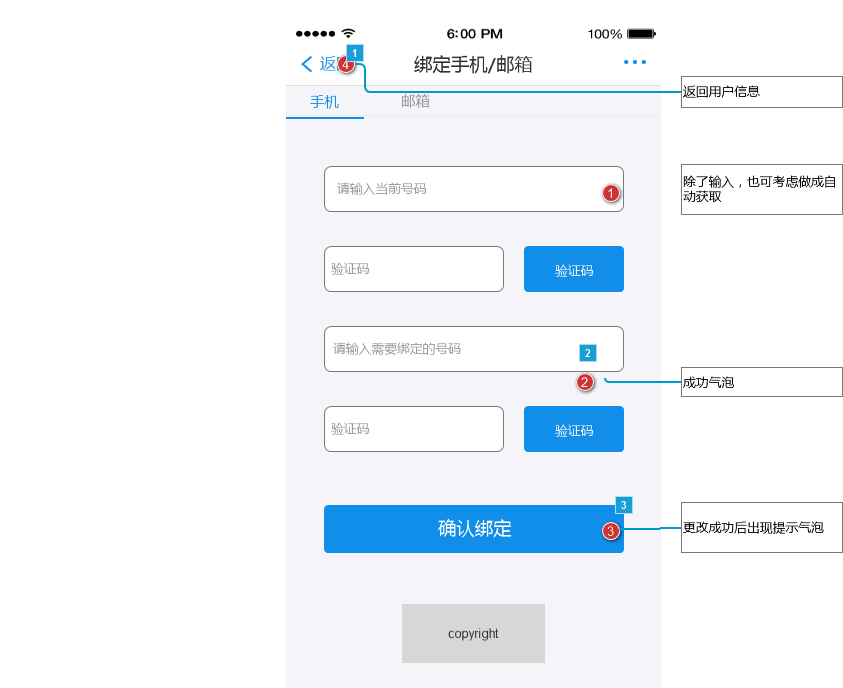
优先级：中

业务流程：无

输入/前置条件：无

页面逻辑

界面原型：



交互：用户点击“关联手机”/“关联邮箱”，如果没有关联过，显示的是“关联手机”，否则显示“更改关联手机”/“更改关联邮箱”。 接着提示进行身份认证，需要输入密码，点击“下一步”。接着输入相关手机号/邮箱地址，获取验证码，通过验证后完成关联。

字段:用户需要输入密码，手机号/邮箱地址，验证码

输出/后置条件：如果有某一个数据输入不正确，应提示错误；关联完成后应提示“关联完成”。

版本

### 修改密码

用户场景：该功能在设置中的账号安全里。用户进入该功能后，首先输入原来的密码，然后输入新密码，确认新密码，完成更改。

优先级：中

输入/前置条件：当前用户有相应的密码

页面逻辑

界面原型：



交互；用户点击“修改密码”，跳出一个输入框，提示输入当前密码，点击“确定”。密码正确的话跳转到新页面，提示输入新密码，下面是再次输入新密码，二者相同则通过验证，完成更改。

字段：用户密码使用加密存储。

输出/后置条件：无

版本

# 非功能需求

## 观感

界面设计应美观简洁，功能分区明确，给用户的第一观感至少不能有压迫感。最好能符合当下相应用户人群的审美需求。

## 易用性

所设计的功能应该使用户能够快速上手，并且归纳合理，尽量避免无关的功能放在一起，而相关的功能相距甚远，并且将常用的功能放在表面上，减少用户掘地三尺找功能的几率。

## 安全性

应用应保证用户的隐私安全，信息安全，避免出现安全漏洞，使不法分子有可乘之机。

## 可维护性

应用应该有较好的可维护性，万一出现了漏洞，或者有什么需要完善的地方，要能够相对轻松地完成更新，而不需要整体推翻重做。