<到云>后台管理系统需求说明书

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 | 2020.03.04 | 制定一个粗略的后台管理需求分析 |
| V02 | 连思璜 | 2020.03.10 | 增加数据字典 |
| V03 | 连思璜 | 2020/3/18 | 增加管理员功能 |
|  |  |  |  |

目录

[1. 引言 1](#_Toc34777515)

[1.1. 目的 1](#_Toc34777516)

[1.2. 背景 1](#_Toc34777517)

[1.3. 定义 1](#_Toc34777518)

[2. 项目概述 2](#_Toc34777519)

[2.1. 产品结构图 2](#_Toc34777520)

[2.2. 产品信息结构图 3](#_Toc34777521)

[2.3. 用户 3](#_Toc34777522)

[2.4. 业务流程 4](#_Toc34777523)

[2.5. 全局说明 4](#_Toc34777524)

[3. 功能详细需求 5](#_Toc34777525)

[3.1. 主页 5](#_Toc34777526)

[3.1.1. 用户信息 5](#_Toc34777527)

[3.1.2. 查看消息 5](#_Toc34777528)

[3.1.3. 修改密码 6](#_Toc34777529)

[3.2. 系统 7](#_Toc34777530)

[3.2.1. 用户权限 7](#_Toc34777531)

[3.2.2. 管理员 8](#_Toc34777532)

[3.2.3. 系统管理 10](#_Toc34777533)

[3.2.4. 消息发布 12](#_Toc34777534)

[3.2.5. 数据字典 13](#_Toc34777535)

[3.3. 用户管理 16](#_Toc34777536)

[3.3.1. 教师/学生用户管理 16](#_Toc34777537)

[3.4. 课程管理 17](#_Toc34777538)

[3.4.1. 所有课程 17](#_Toc34777539)

[3.4.2. 我的课程 19](#_Toc34777540)

[3.5. 班级管理 21](#_Toc34777541)

[3.5.1. 班级信息 21](#_Toc34777542)

[3.5.2. 学生信息 23](#_Toc34777543)

[3.5.3. 签到情况 24](#_Toc34777544)

[4. 非功能需求 26](#_Toc34777545)

[4.1. 观感 26](#_Toc34777546)

[4.2. 易用性 26](#_Toc34777547)

[4.3. 安全性 26](#_Toc34777548)

[4.4. 可维护性 26](#_Toc34777549)

# 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 |
| 主页 | 后台管理系统主页 | 显示用户信息、系统消息 |
| 系统 | 用户权限 | 给管理员和教师分配不同的权限 |
| 新增管理员 |  |
| 消息发布 | 管理员发布系统消息 |
| 系统管理 | 管理隐私政策、关于我们等文档 |
| 用户管理 | 教师用户管理 | 可以查看用户信息、删除用户 |
| 学生用户管理 | 可以查看用户信息、删除用户 |
| 班级管理 | 学生管理 | 查看每个班级对应的学生 |
| 查看签到 | 查看签到信息 |
| 课程管理 | 课程管理 | 管理员可以看到所有课程、教师可以看到自己创建的课程，可以有增删操作 |

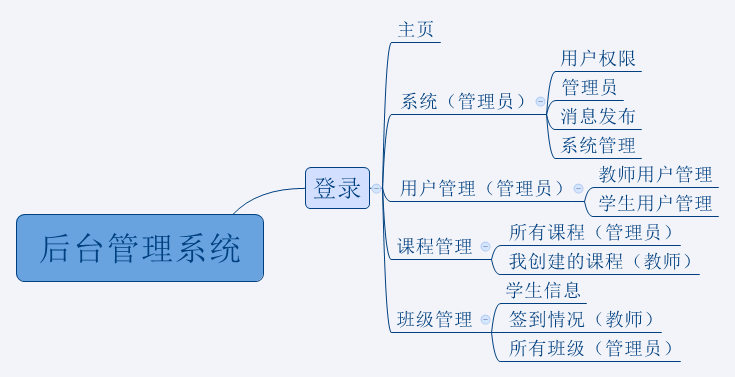


Fig 1.产品结构图

## 产品信息结构图

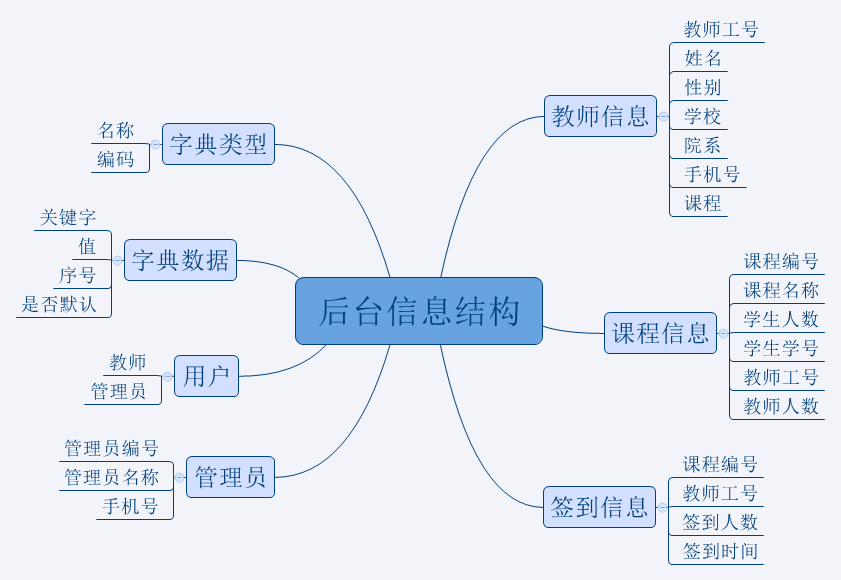


Fig 2.产品信息结构图

## 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、查看签到信息、查看班级信息 |  |
| 管理员 | 管理系统权限、 | 管理员具有最高权限 |



Fig 3.教师管理员用例图

## 业务流程

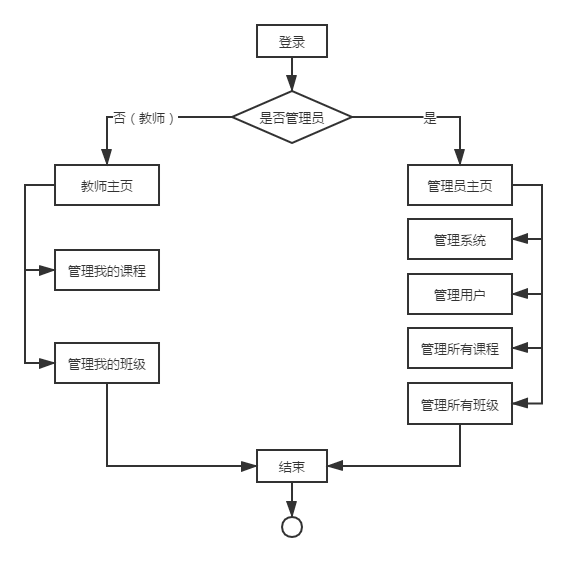


Fig 5.业务流程图

## 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。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。点击“删除”时，需要有弹窗跳出，询问是否“确认删除？”。

# 功能详细需求

## 主页

### 用户信息

用户场景：用户登录后，进入该系统的页面。在这里可以看到用户的信息，例如昵称、头像等等。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。

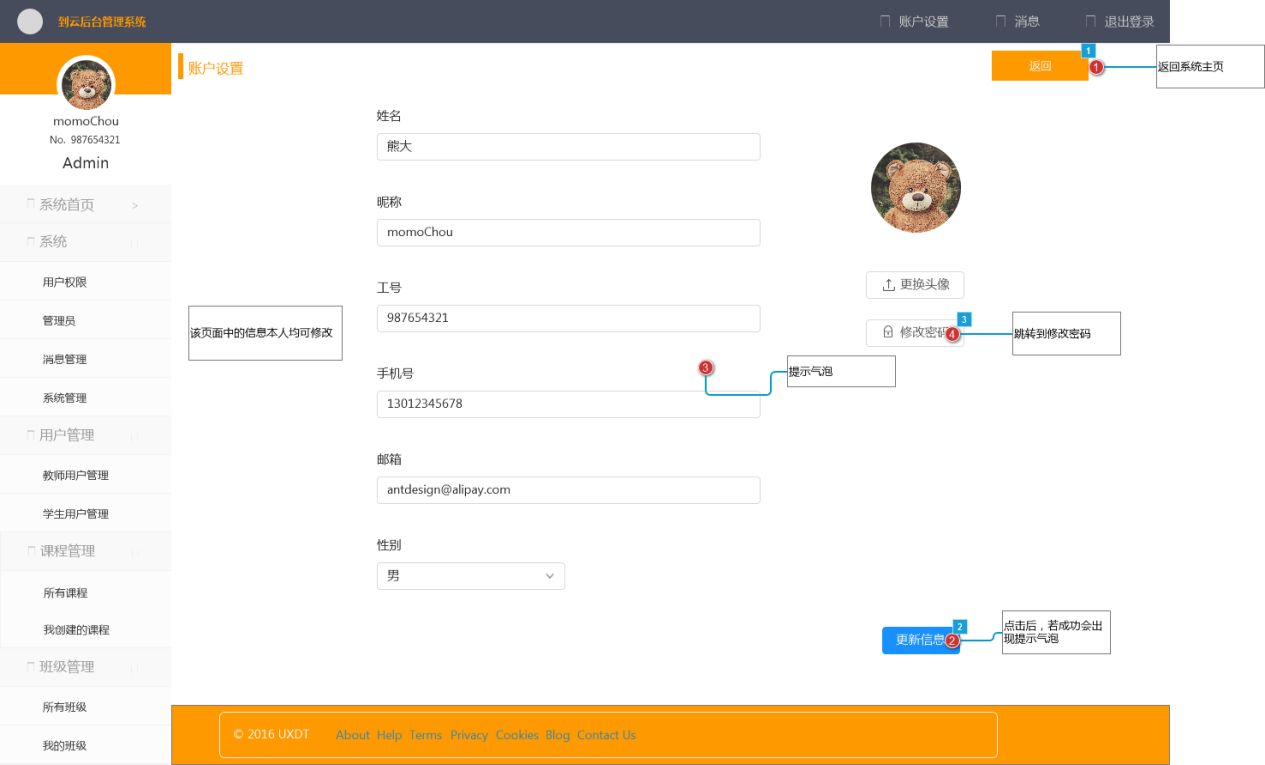
优先级：高

业务流程：无

输入/前置条件：用户要先注册

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”。

### 查看消息

用户场景：在进入主页后，用户可以看到管理员发布的相关系统信息

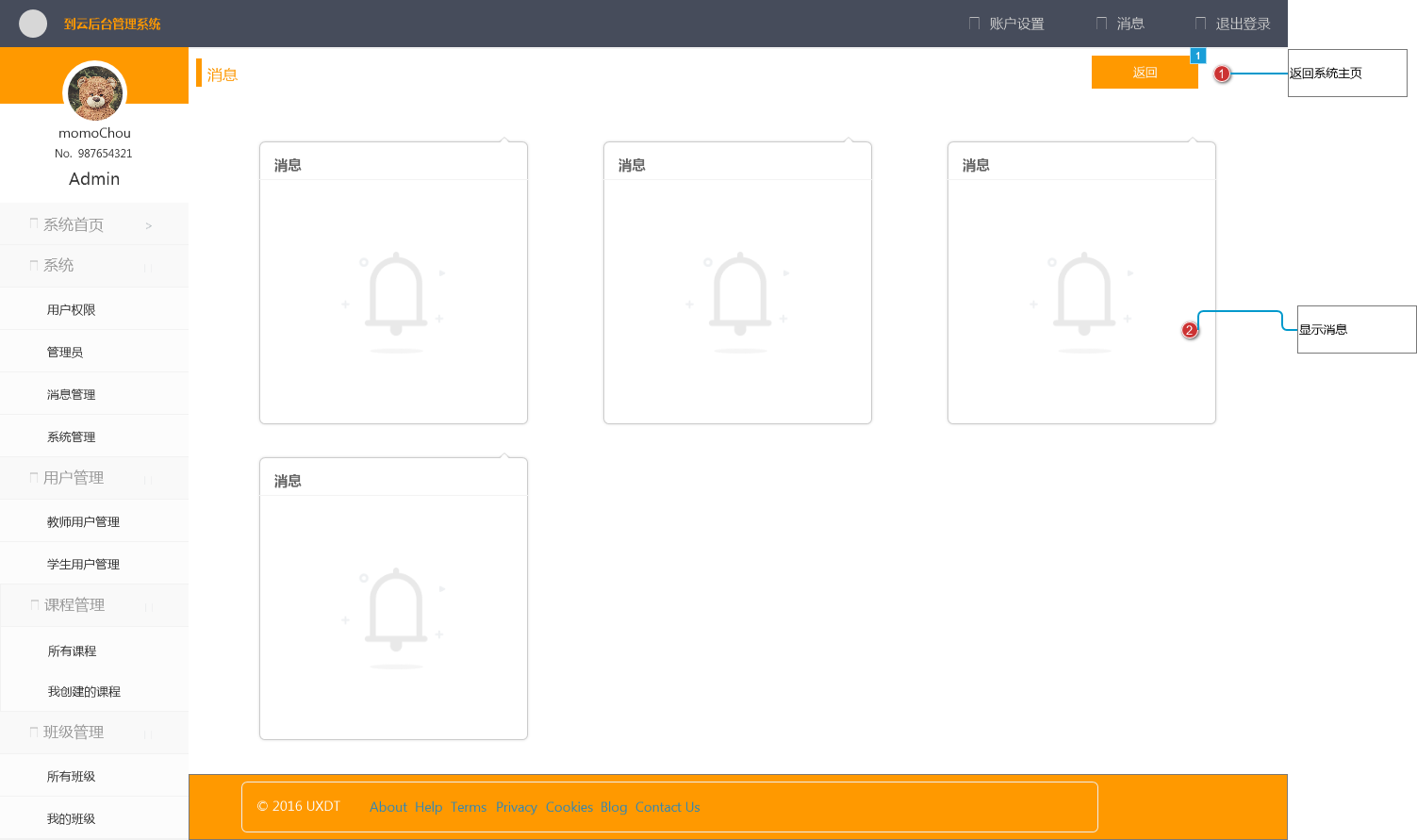
优先级：中

业务流程：无

输入/前置条件：管理员用户发布系统消息

页面逻辑

界面原型：



交互：

字段：管理员发布的消息存储在数据库中，页面展示的信息从数据库中读取

输出/后置条件：

### 修改密码

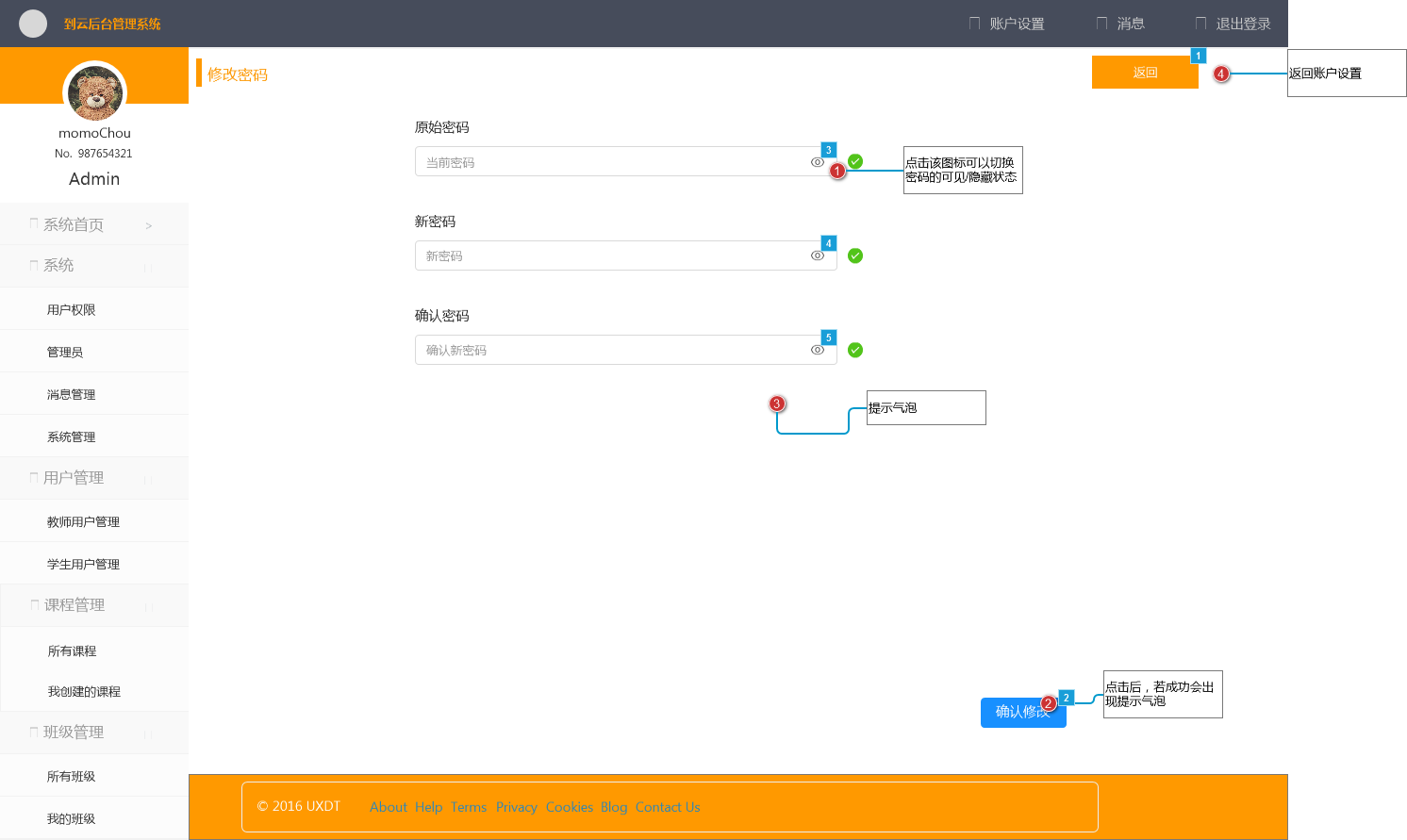
用户场景：该功能在用户信息里。用户进入该功能后，首先输入原来的密码，然后输入新密码，确认新密码，完成更改。

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

## 系统

### 用户权限

用户场景：管理员从菜单中进入此功能，对教师用户和管理员用户的权限进行管理（一般管理员拥有全部的权限）

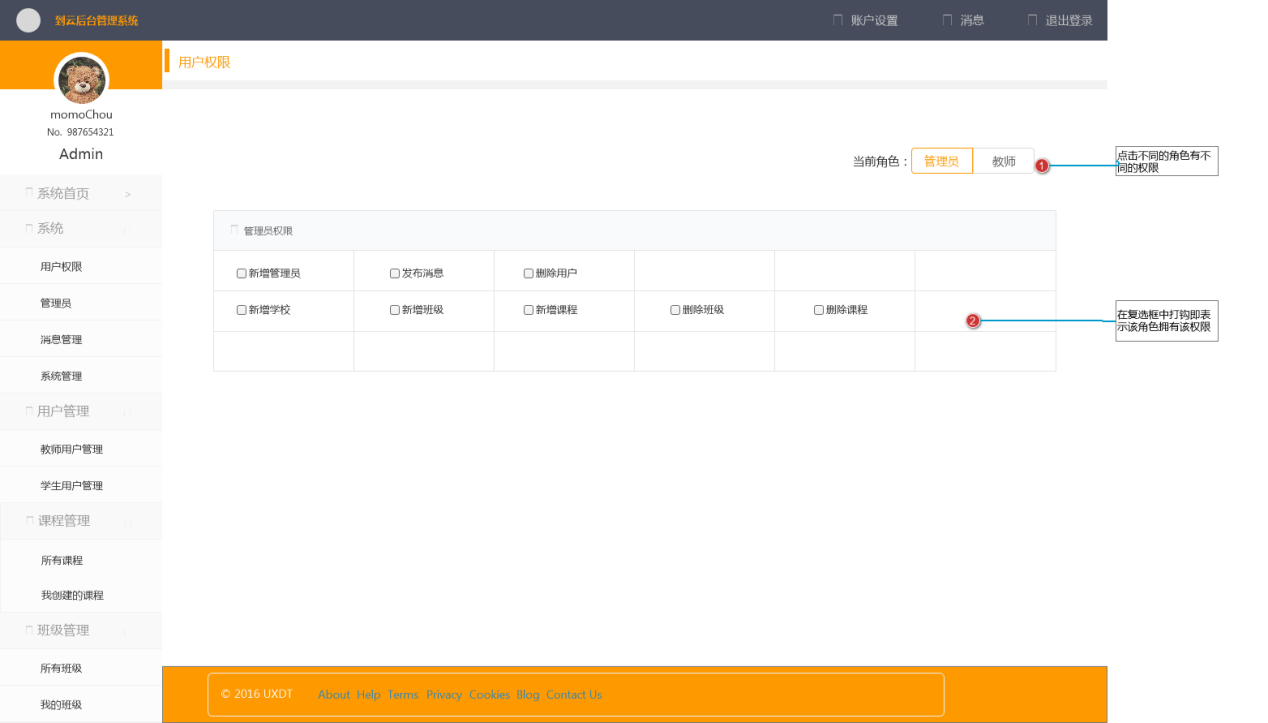
优先级：高

业务流程：无

输入/前置条件：操作用户是管理员

页面逻辑

界面原型：



交互：管理员在角色相应的权限选择框中打钩，即表示开通权限；否则就是没开通该权限。

字段：

输出/后置条件：若权限开通，用户可以进行相应的操作；反之，用户无法进行相应的操作

### 管理员

用户场景：管理员从菜单中进入此功能，点击“新增”按钮添加管理员，在相应的管理员条目中点击“删除”删除管理员

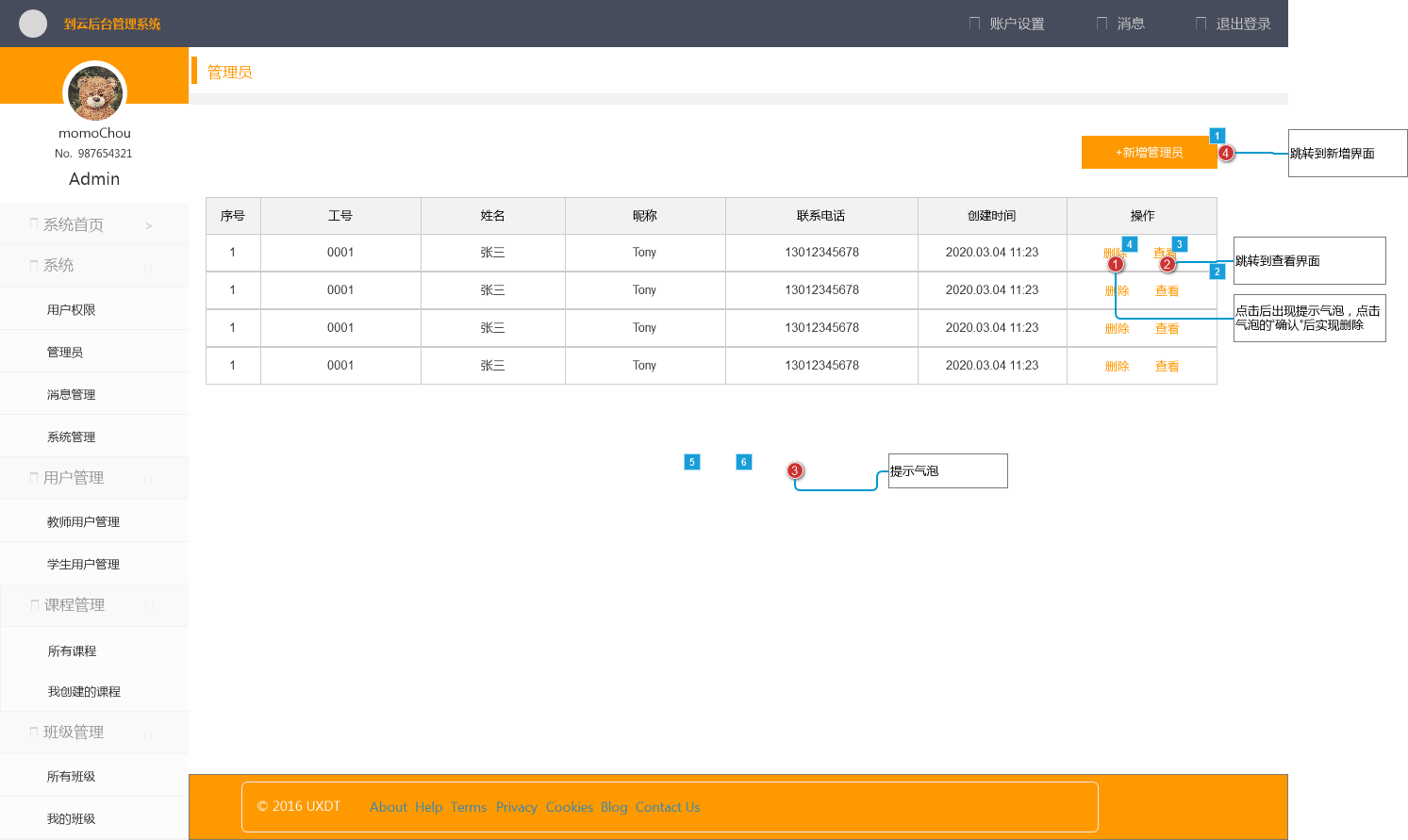
优先级：高

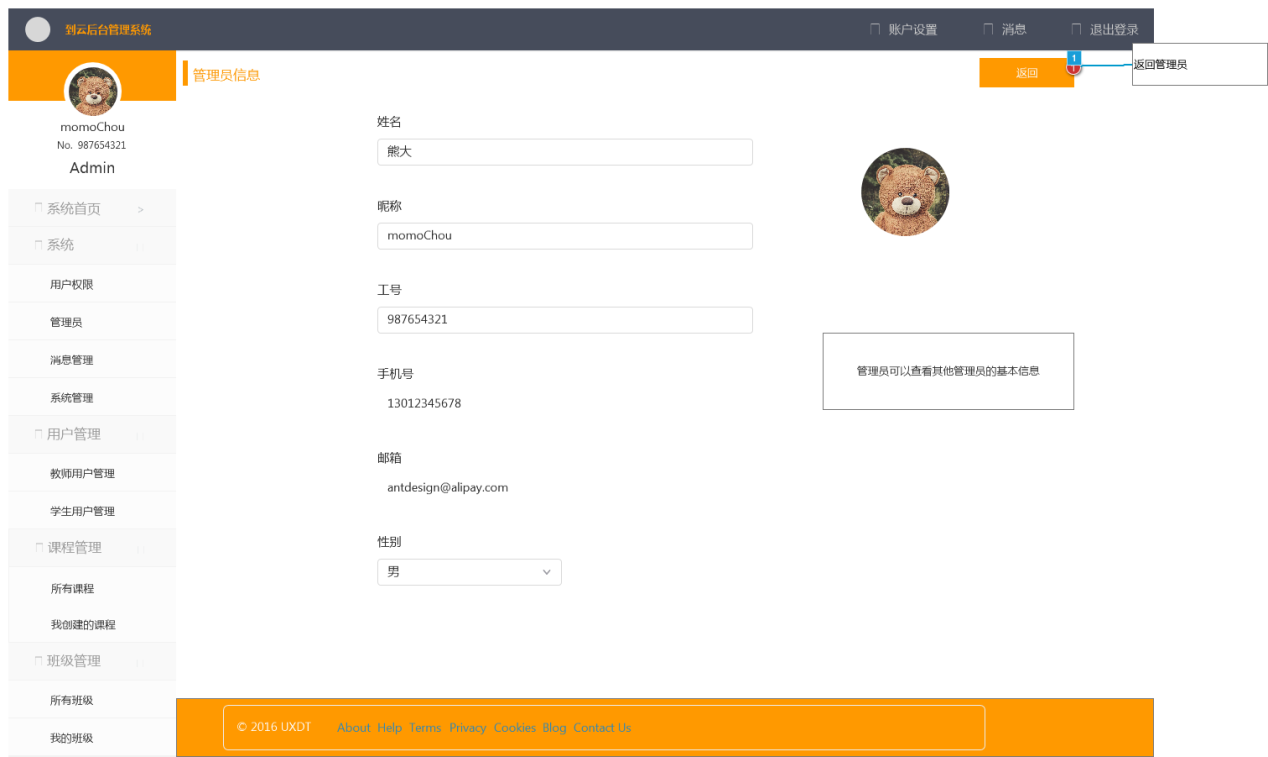
业务流程：无

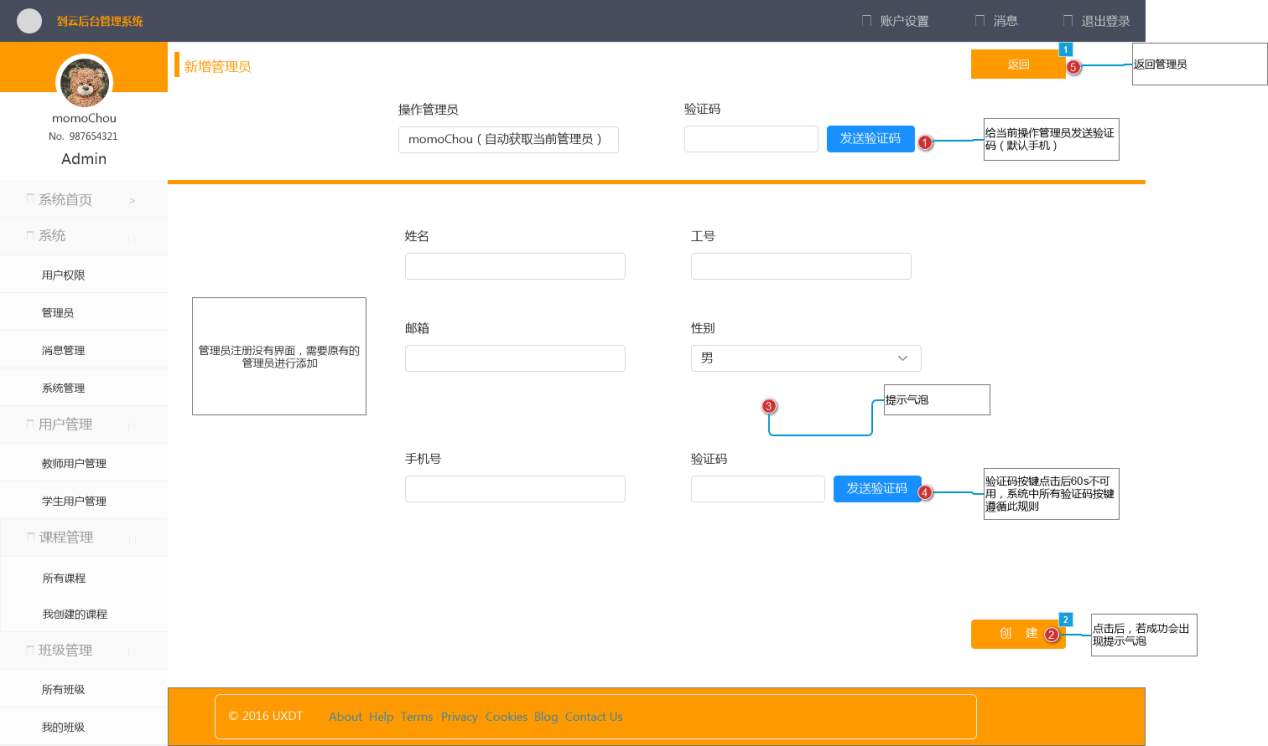
输入/前置条件：操作用户是管理员

页面逻辑

界面原型：







交互：管理员新增管理员时，和注册类似；删除管理员时，需要有弹窗跳出，提示是否确认删除，以防误删。

字段： 新增的管理员个人信息会保存到数据库中；删除的管理员虽然被删除了，但是信息仍然留在数据库中，只是变成了不可操作。

输出/后置条件：新管理员被添加之后，就可以进行所有管理员操作了。

### 系统管理

用户场景：管理员从菜单中进入此功能，管理员可以在这里对一些文档进行维护，增加与系统运行相关的数据，例如考勤签到的默认经验值。

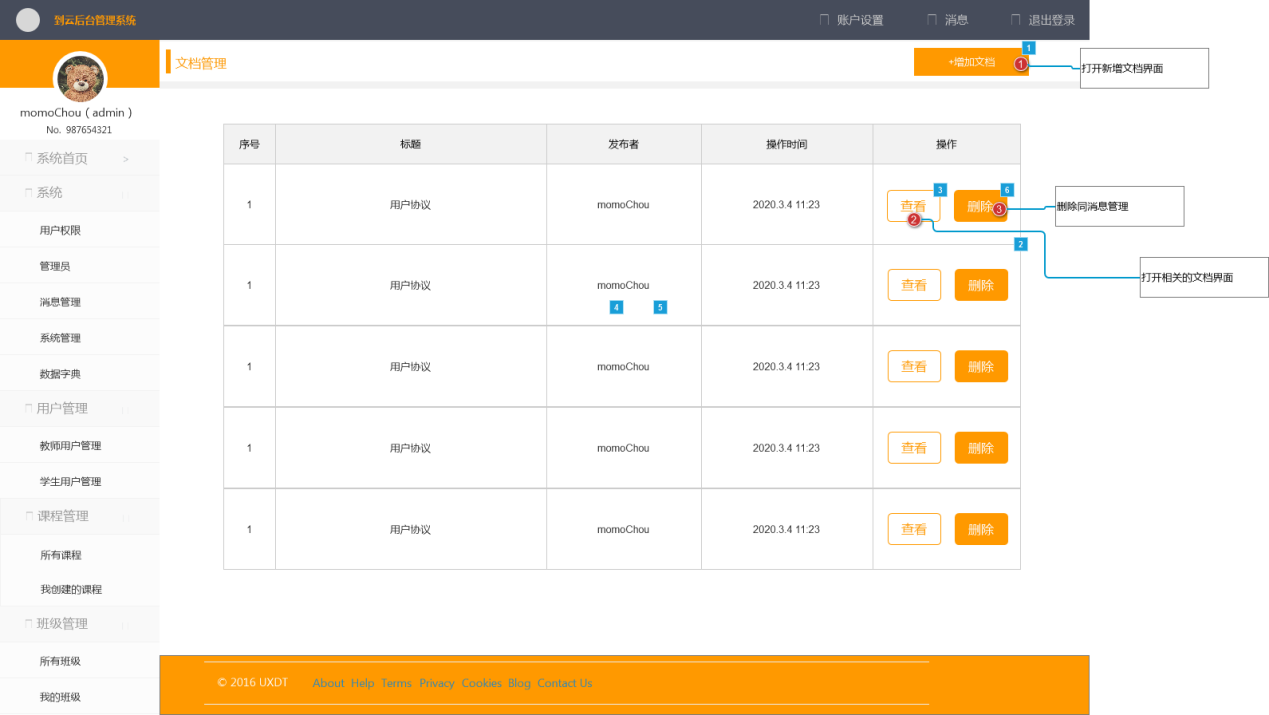
优先级：高

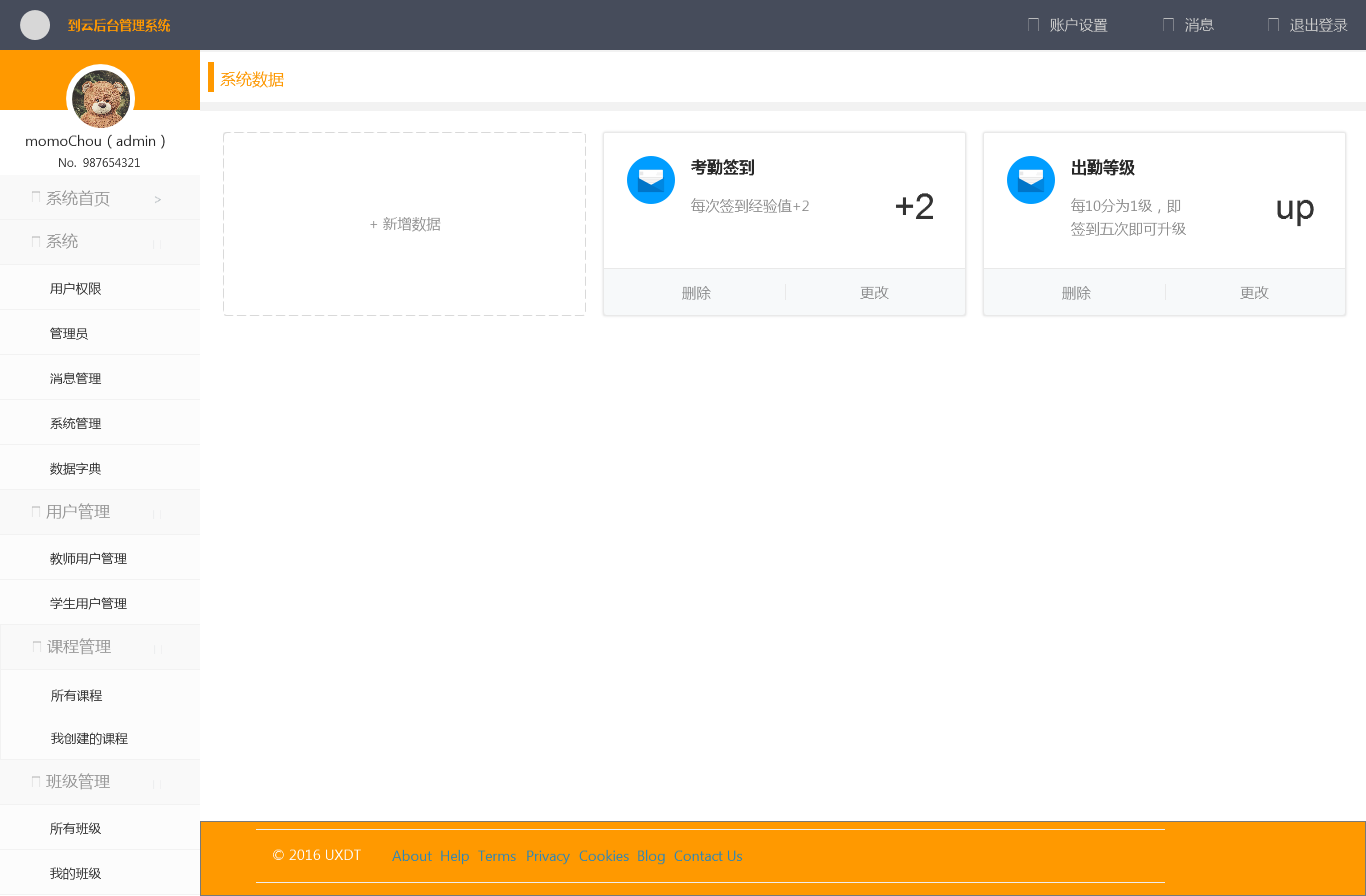
业务流程：无

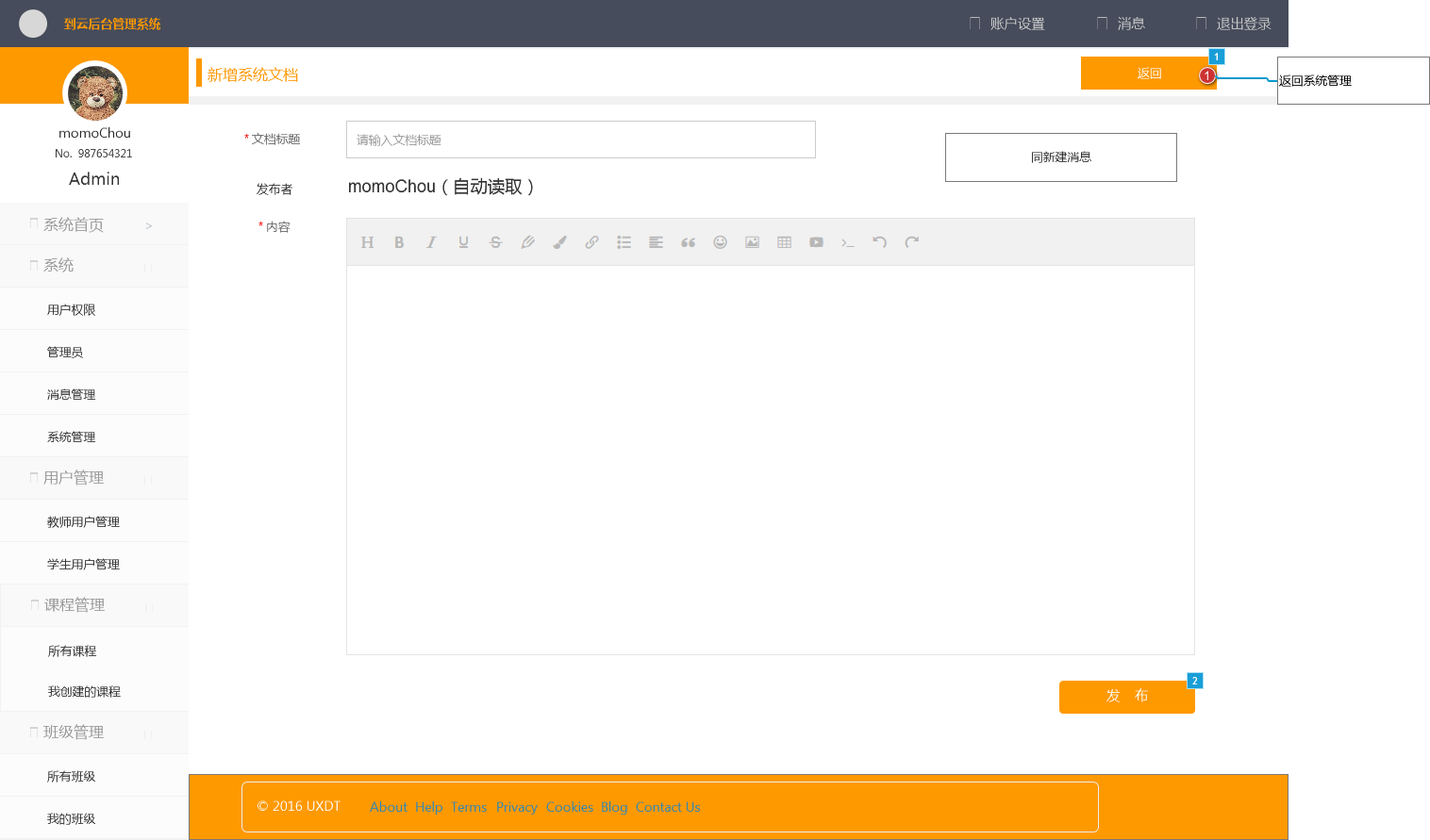
输入/前置条件：操作用户是管理员

页面逻辑

界面原型：







交互：界面上有一个文本框，可以适配移动端，管理员可以在此进行文档的编辑，编辑结束点击“完成”，更改后的文档可以同步更新到每个用户端。

字段： 管理员编辑的文档储在数据库中

输出/后置条件：管理员发布之后，其他用户端经过联网后会显示出来

### 消息发布

用户场景：管理员从菜单中进入此功能，管理员可以在这里将一些更新、维护等消息发送给其他系统用户。

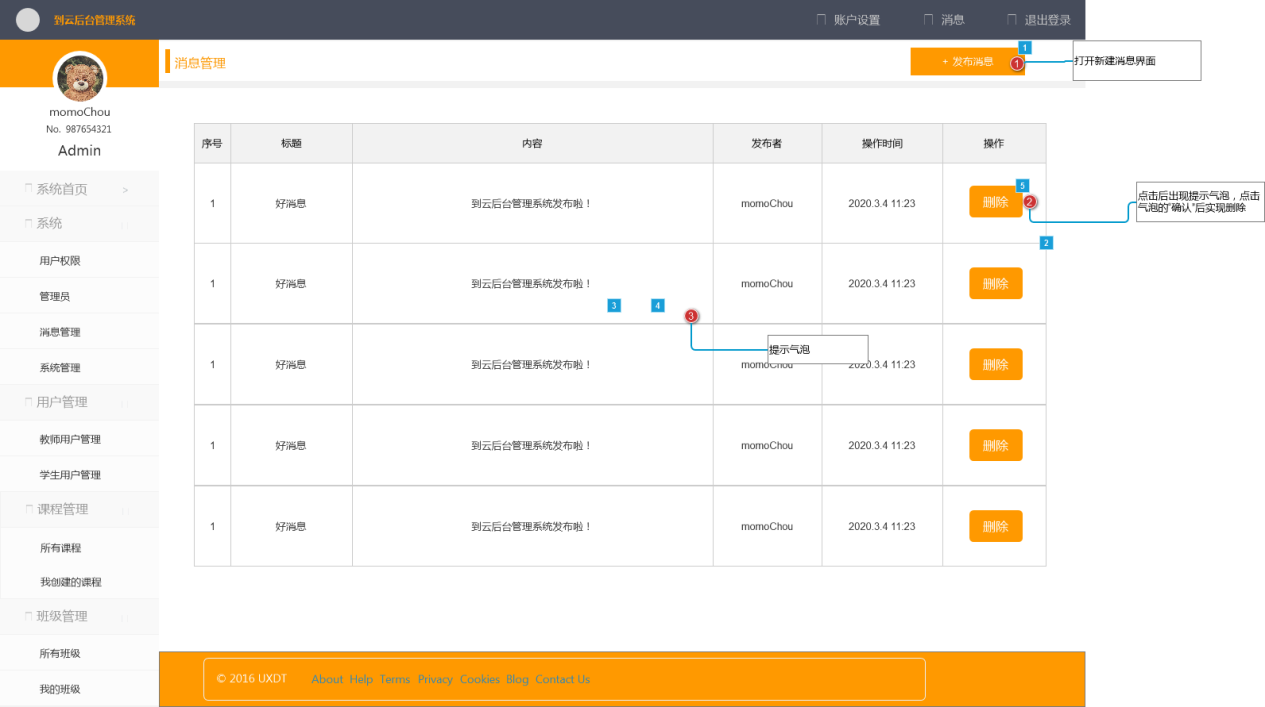
优先级：高

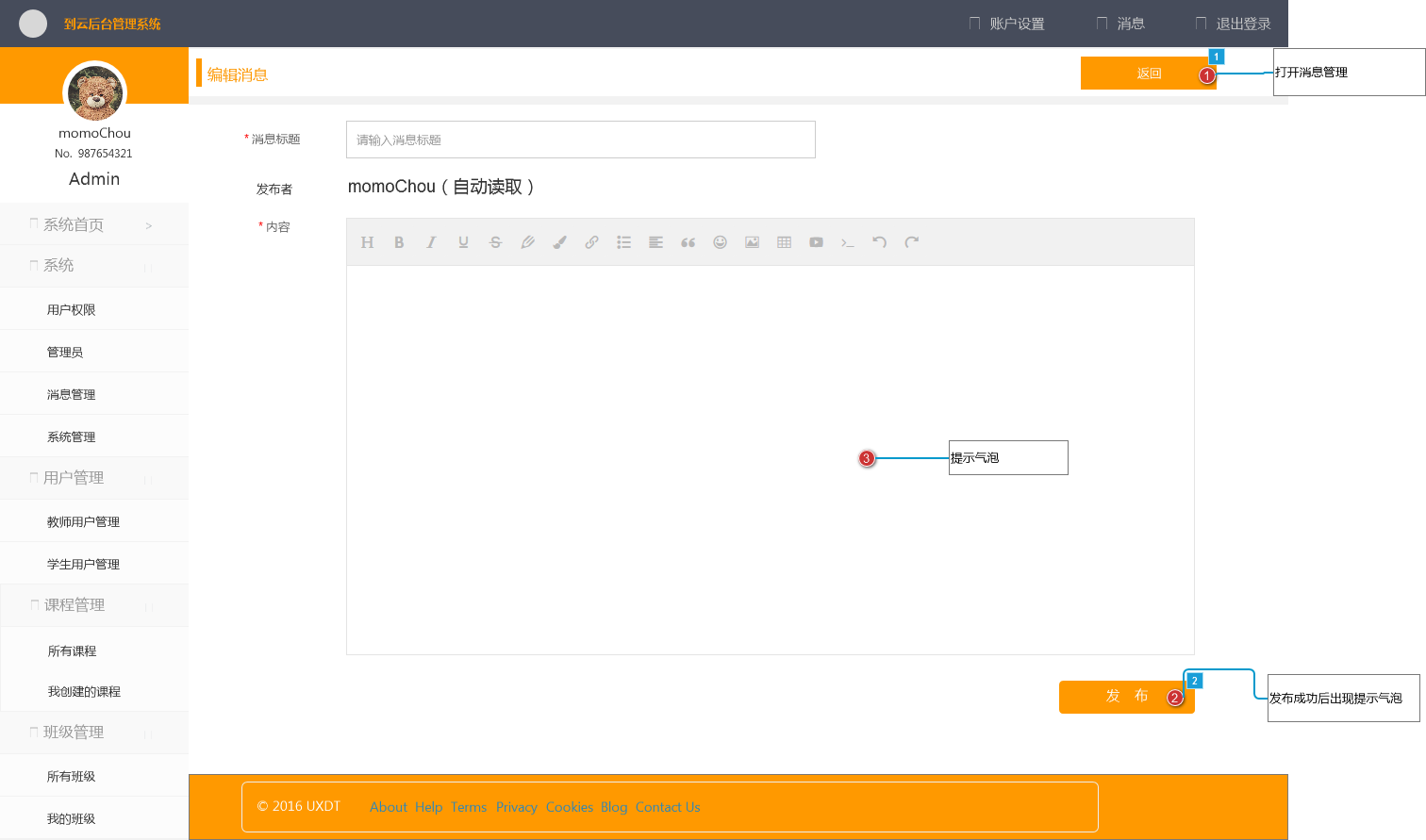
业务流程：无

输入/前置条件：操作用户是管理员

页面逻辑

界面原型：





交互：管理员发布消息的界面类似邮箱中的发邮件，界面中有两个文本框，上面的是标题，下面的是内容输入框。管理员编辑后，点击发布，其他用户就能在主页的消息框中阅览。

字段： 管理员发出的消息会被存储在数据库中

输出/后置条件：管理员发布之后，其他用户的主页就会将其显示出来。

### 数据字典

用户场景：管理员从菜单中“系统”的下拉列表进入此功能，对系统中需要选择的数据进行管理。

优先级：高

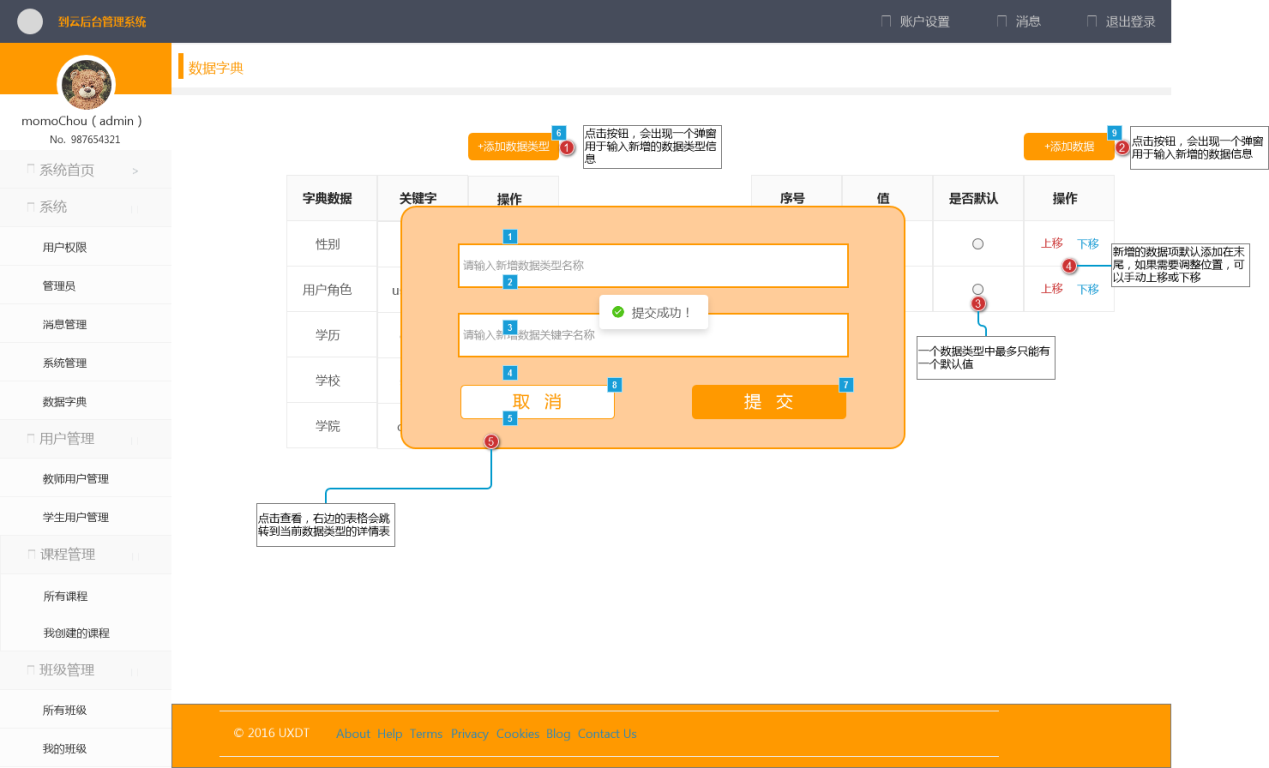
业务流程：无

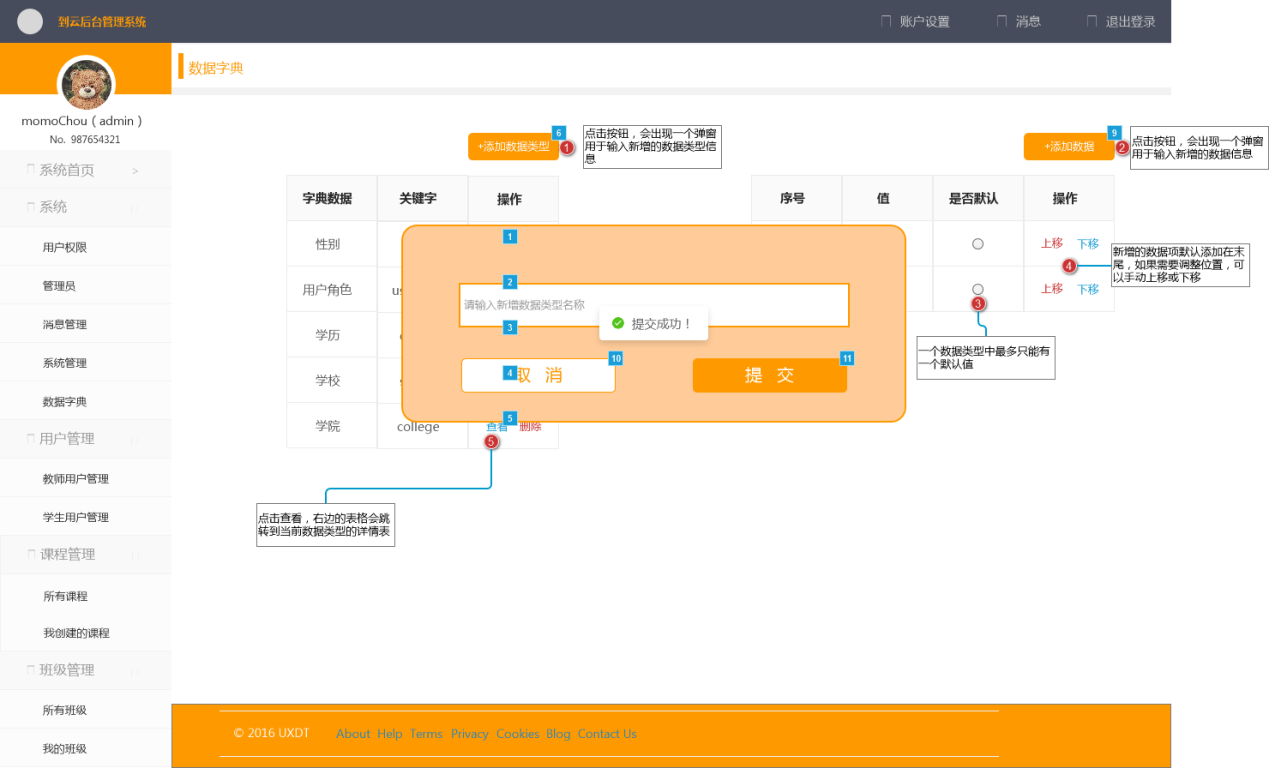
输入/前置条件：操作用户是管理员

页面逻辑

界面原型：







交互：管理员进入页面后可以对已有的数据进行修改，既可以添加新条目，也可以在已有条目中添加新的元素。

字段：管理员所作出的修改会被存储在数据库中。学校、学院需要技术人员在后台添加，示例中仅为举例。

输出/后置条件：管理员发布之后，其他用户的相应页面就会将其显示出来。

示例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字典数据 | 关键字 | 值 | 序号 | 是否默认 |
| 性别 | sex | 男 | 1 | Y |
| 女 | 2 |  |
| 用户类型 | userType | 学生 | 1 | Y |
| 教师 | 2 |  |
| 学历 | eduBG | 高中 | 1 |  |
| 本科 | 2 | Y |
| 硕士 | 3 |  |
| 博士 | 4 |  |
| 学校 | school | 福州大学 | 1 |  |
| 厦门大学 | 2 |  |
| 学院 | college | 数计学院 | 1 |  |
| 物信学院 | 2 |  |

## 用户管理

### 教师/学生用户管理

用户场景：管理员从菜单中进入此功能，管理员可以对注册该系统/应用的老师/学生用户进行管理。

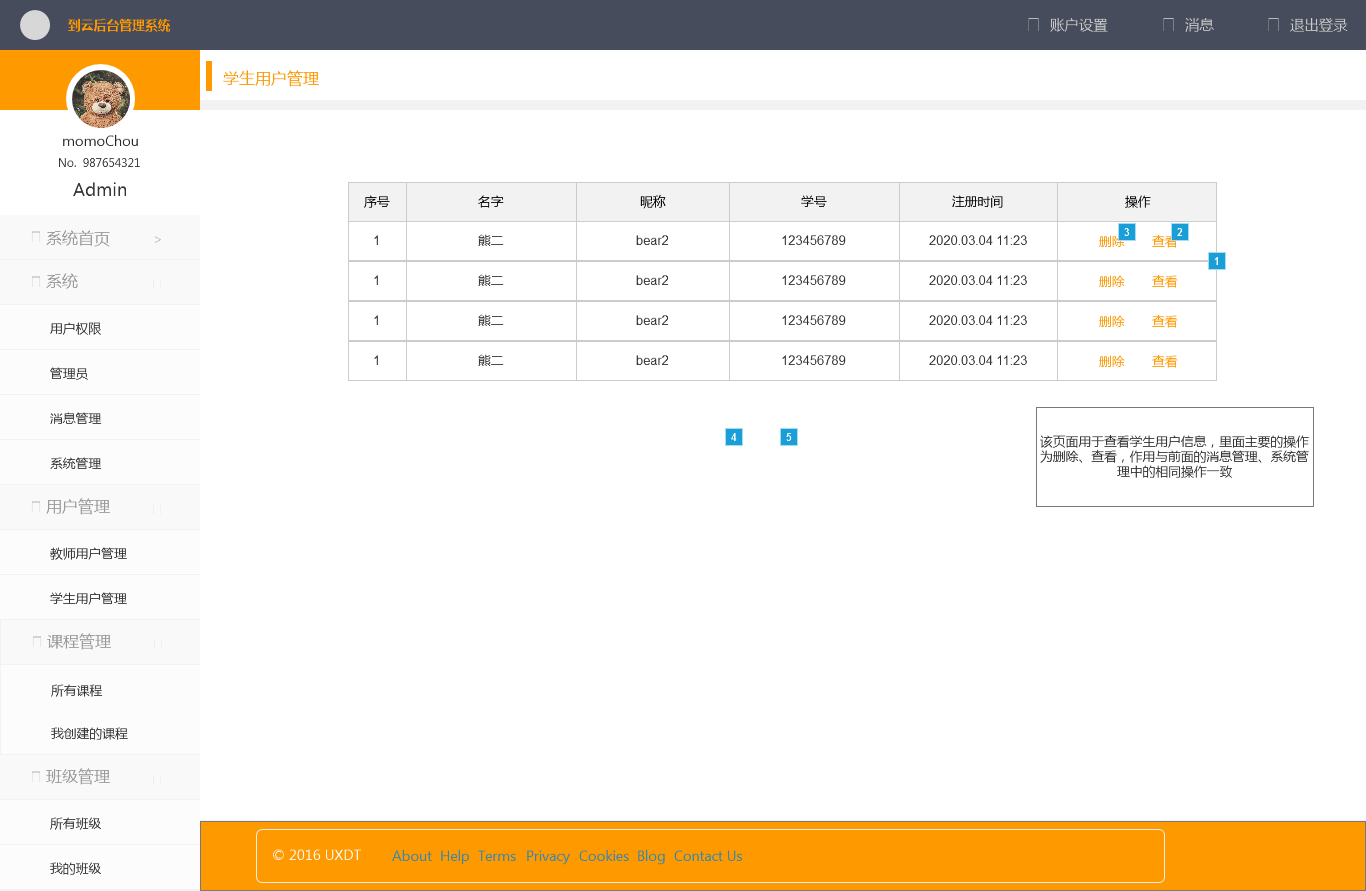
优先级：高

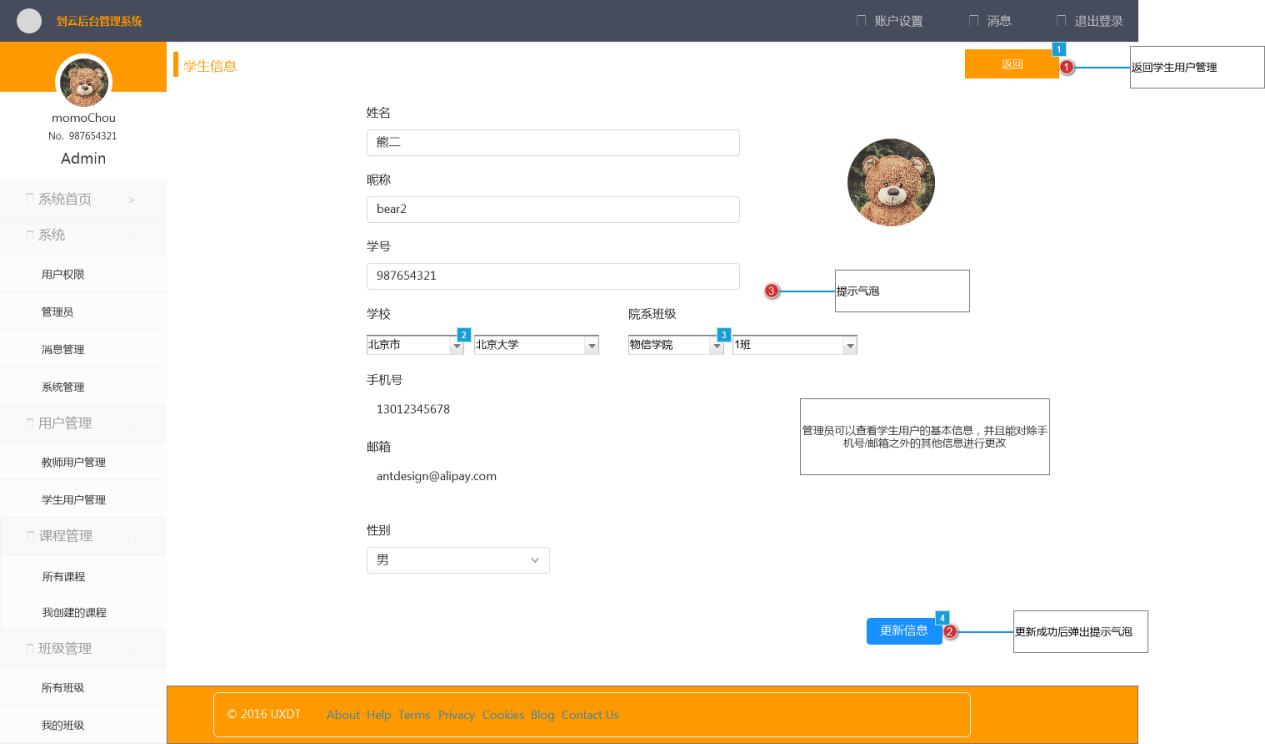
业务流程：无

输入/前置条件：操作用户是管理员，用户存在

页面逻辑

界面原型：





交互：管理员点击每个用户条目中的“查看”，可以查看用户的基本信息；点击“删除”可以删除相应的用户

字段： 用户信息从数据库中读取；管理员若对数据进行操作会被保存到数据库中。

输出/后置条件： 若用户被删除，则数据库中就不存在该用户

## 课程管理

### 所有课程

用户场景：管理员从菜单中的“课程管理”进入此功能，管理员可以对所有已创建的班课进行管理，增删改查均可。

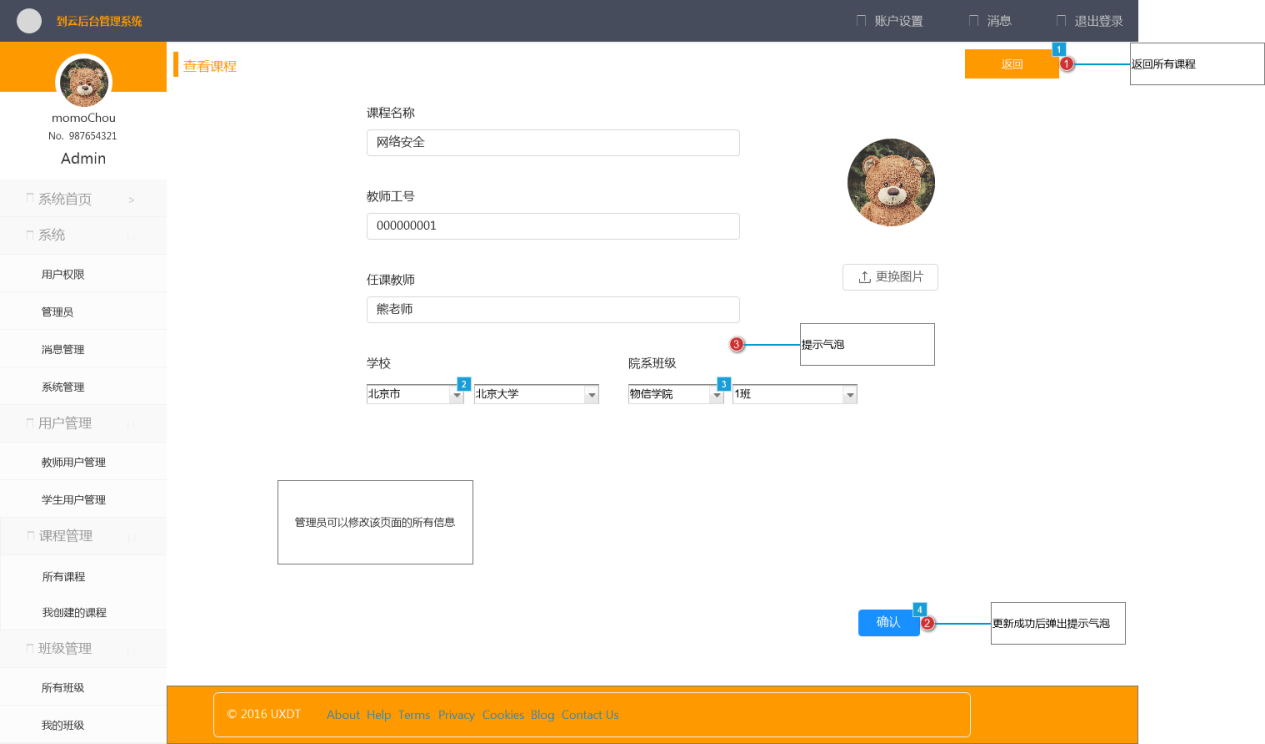
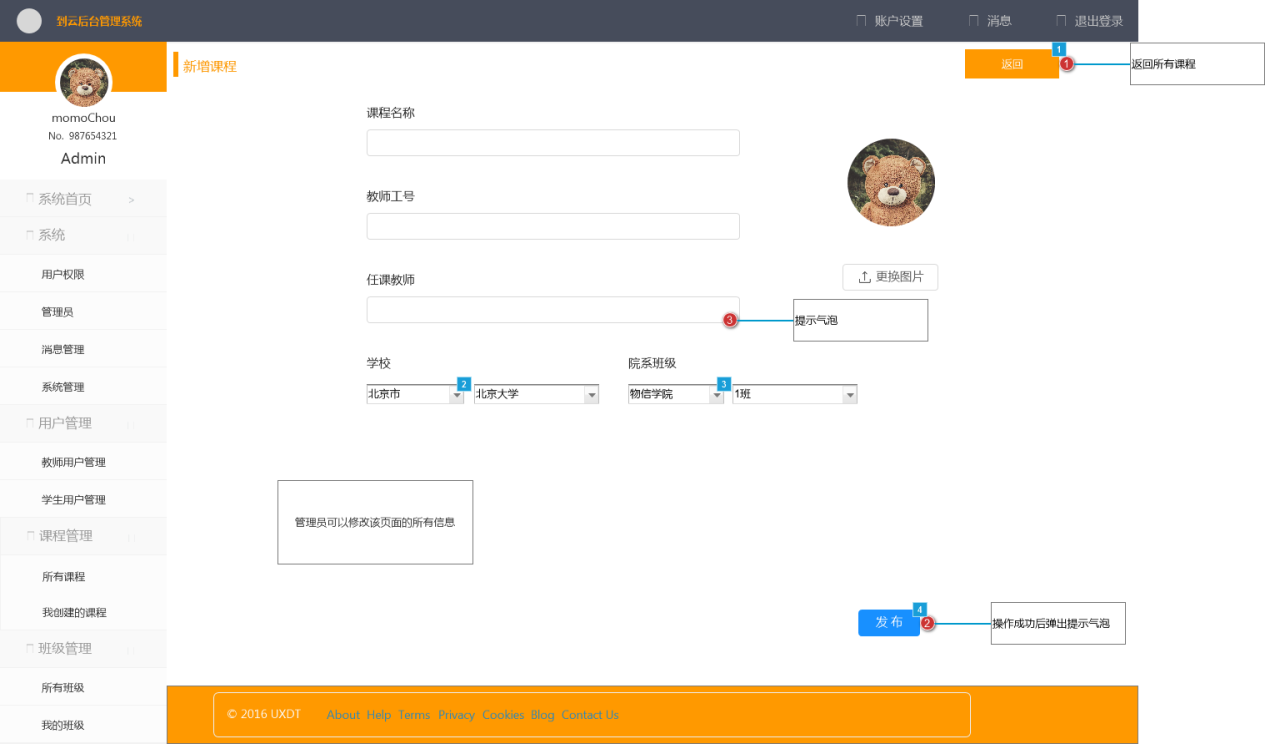
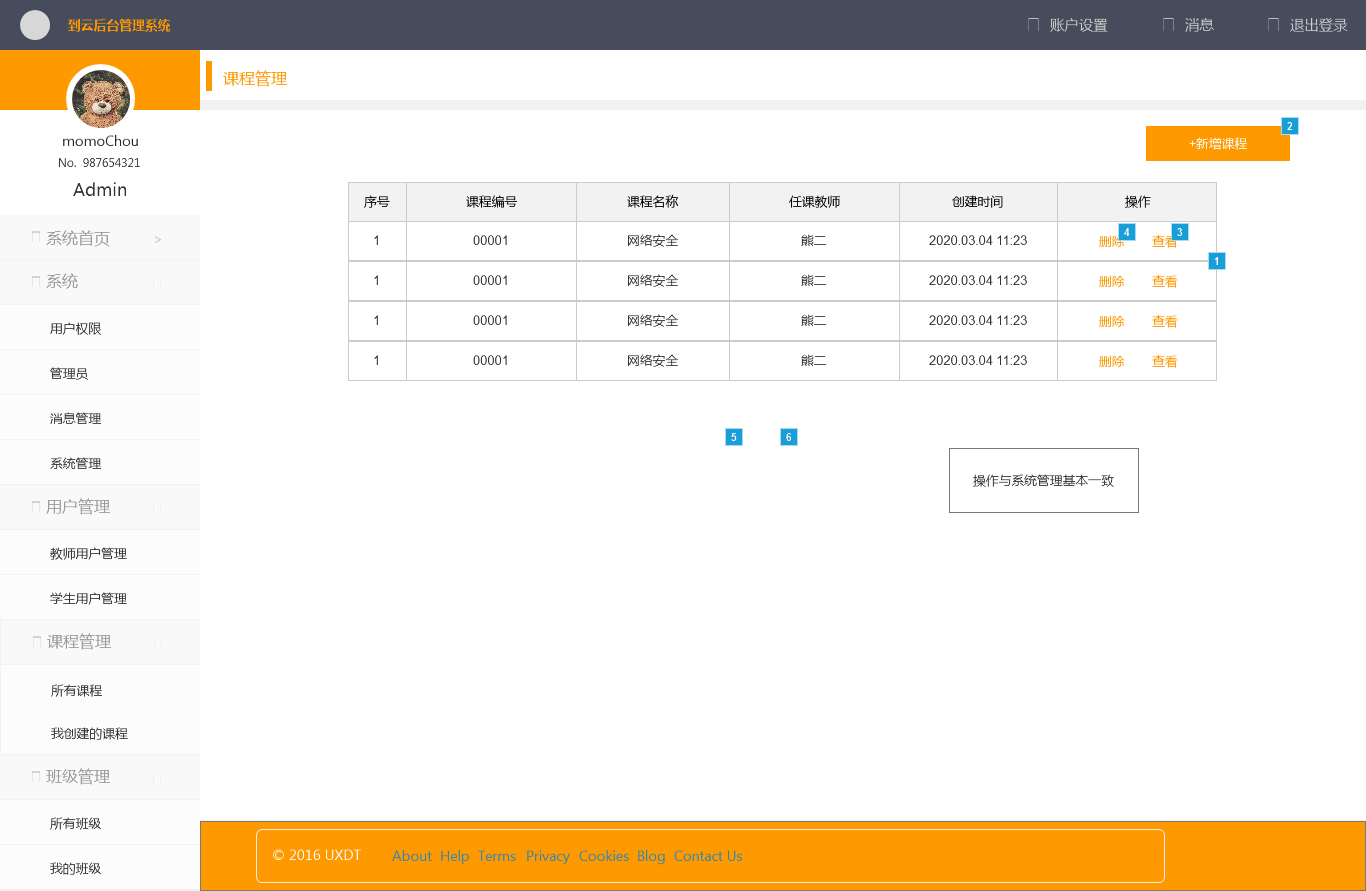
优先级：高

业务流程：无

输入/前置条件：操作用户是管理员，若对课程进行操作，需要课程存在于数据库中

页面逻辑

界面原型：



交互：管理员点击每个课程条目中的“查看”，可以查看课程的基本信息，例如任课教师、上课班级等；点击“删除”可以删除相应的课程；点击“修改”可以修改课程的一些基本信息，例如当变更任课教师时，管理员可以进行更改；点击右上角的“添加”可以创建新的班课。

字段： 课程信息从数据库中读取；管理员若对数据进行操作会被保存到数据库中。

输出/后置条件：

### 我的课程

用户场景：教师从菜单中的“课程管理”进入此功能，教师可以对所有自己创建的班课进行管理，增删改查均可。

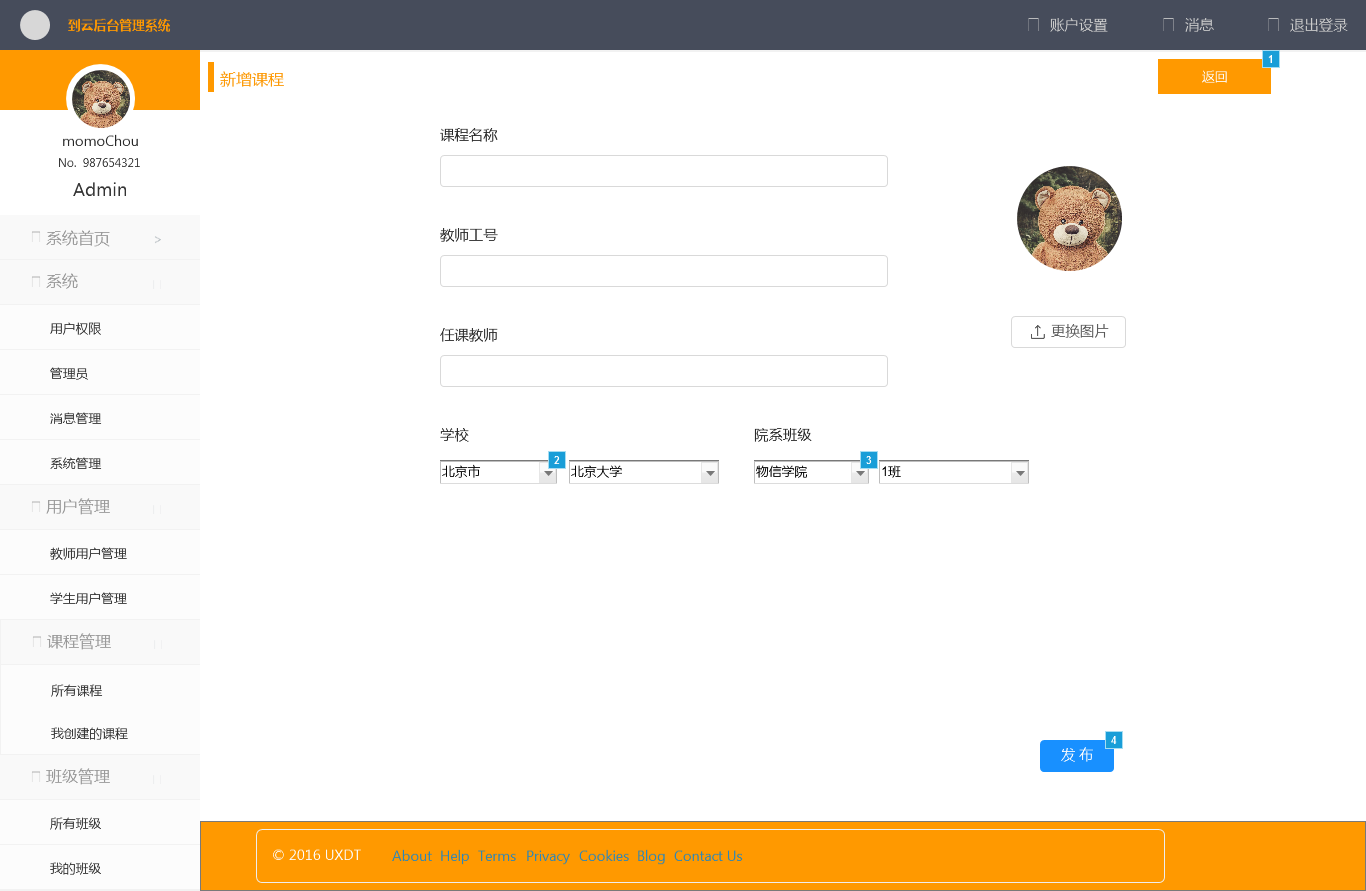
优先级：高

业务流程：无

输入/前置条件：操作用户是教师，若对课程进行操作，需要课程存在于数据库中

页面逻辑

界面原型：

交互：教师点击每个课程条目中的“查看”，可以查看课程的基本信息，例如任课教师、上课班级等；点击“删除”可以删除相应的课程；点击“修改”可以修改课程的一些基本信息，例如当变更上课班级时，教师可以进行更改；点击右上角的“添加”可以创建新的班课。

字段： 课程信息从数据库中读取；教师若对数据进行操作会被保存到数据库中。

输出/后置条件：

## 班级管理

### 班级信息

用户场景：教师从菜单中的“班级管理”进入此功能，教师可以对所有当前自己正在授课的班级进行管理，增删改查均可。

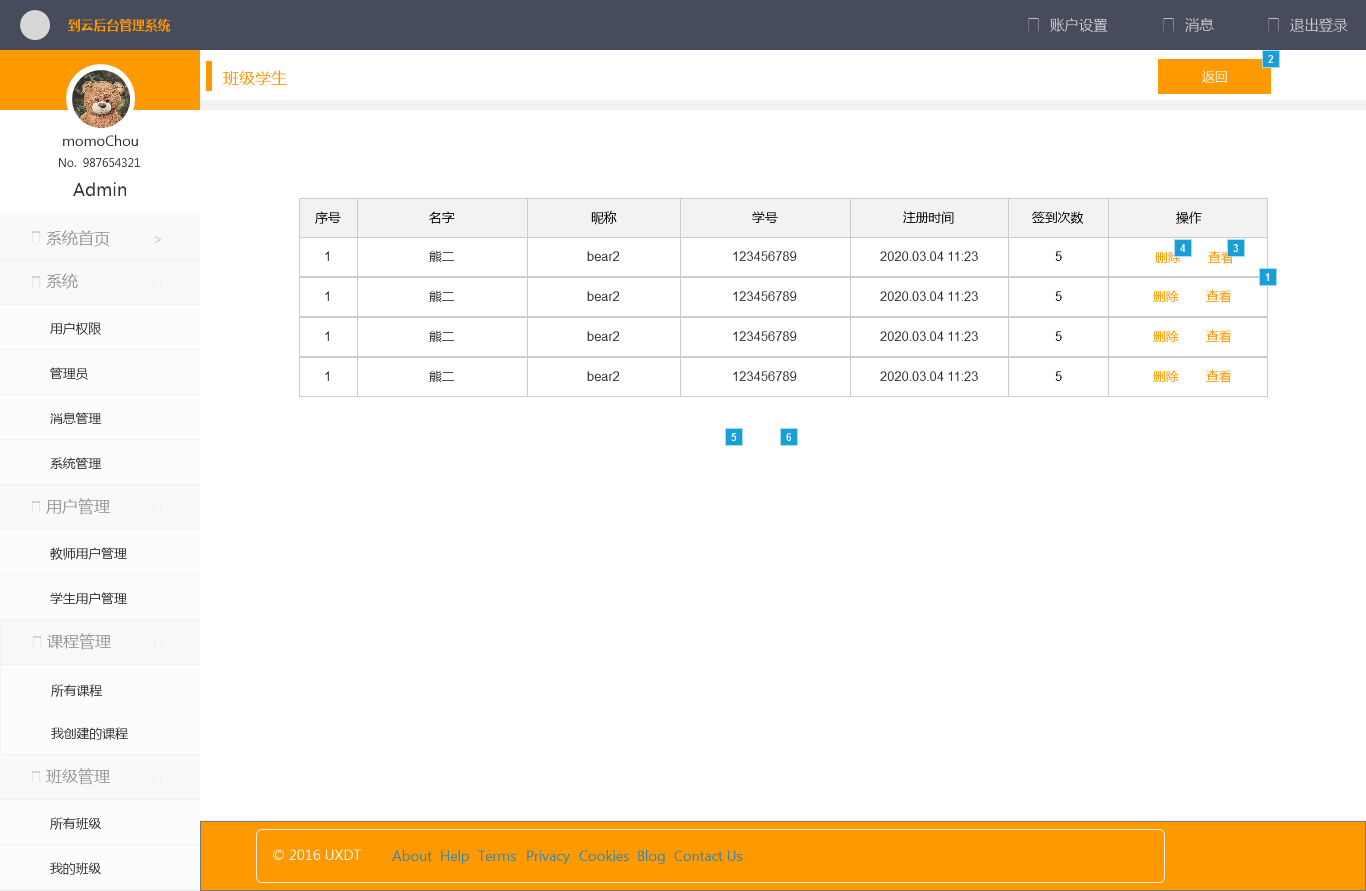
优先级：高

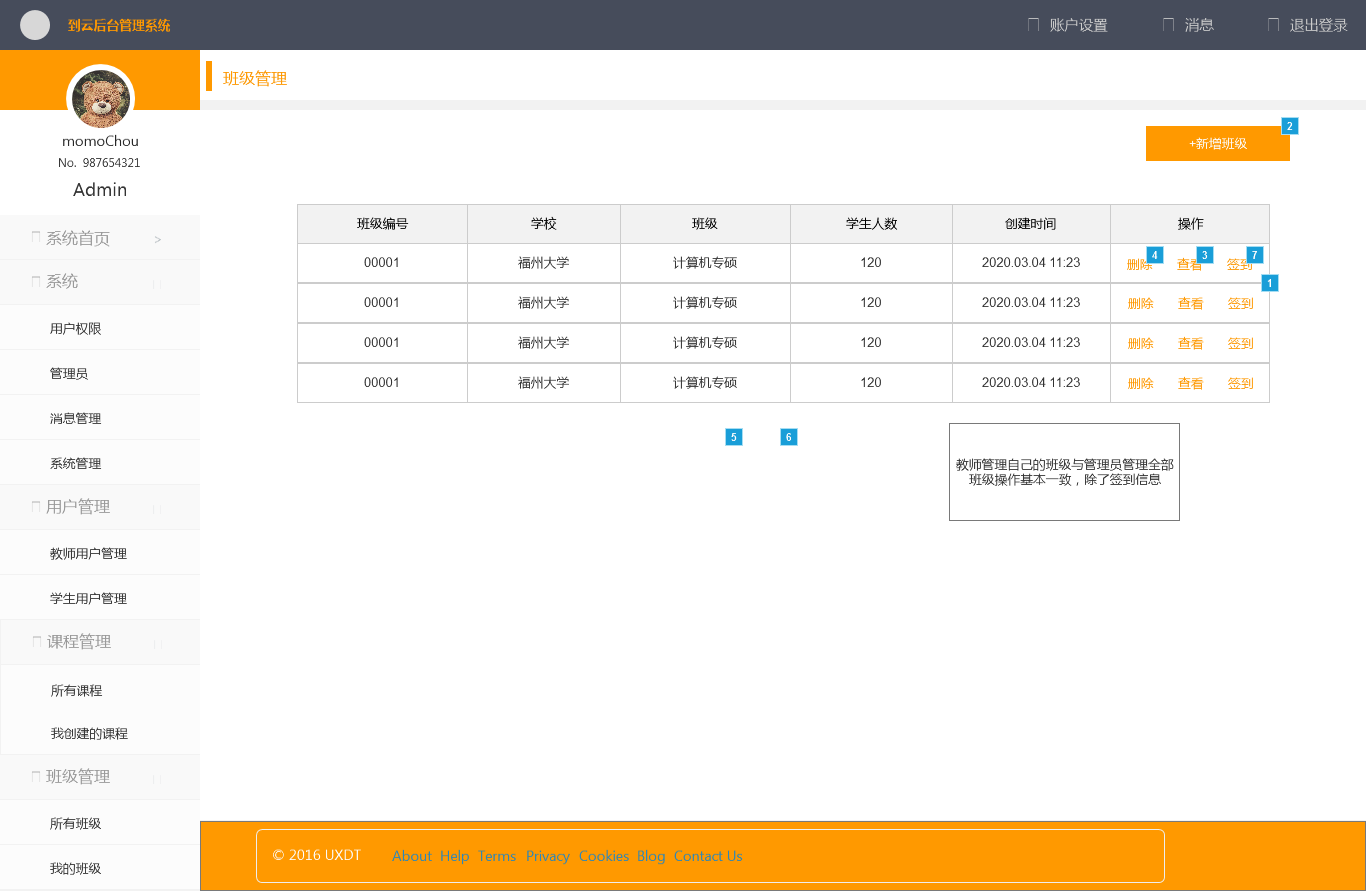
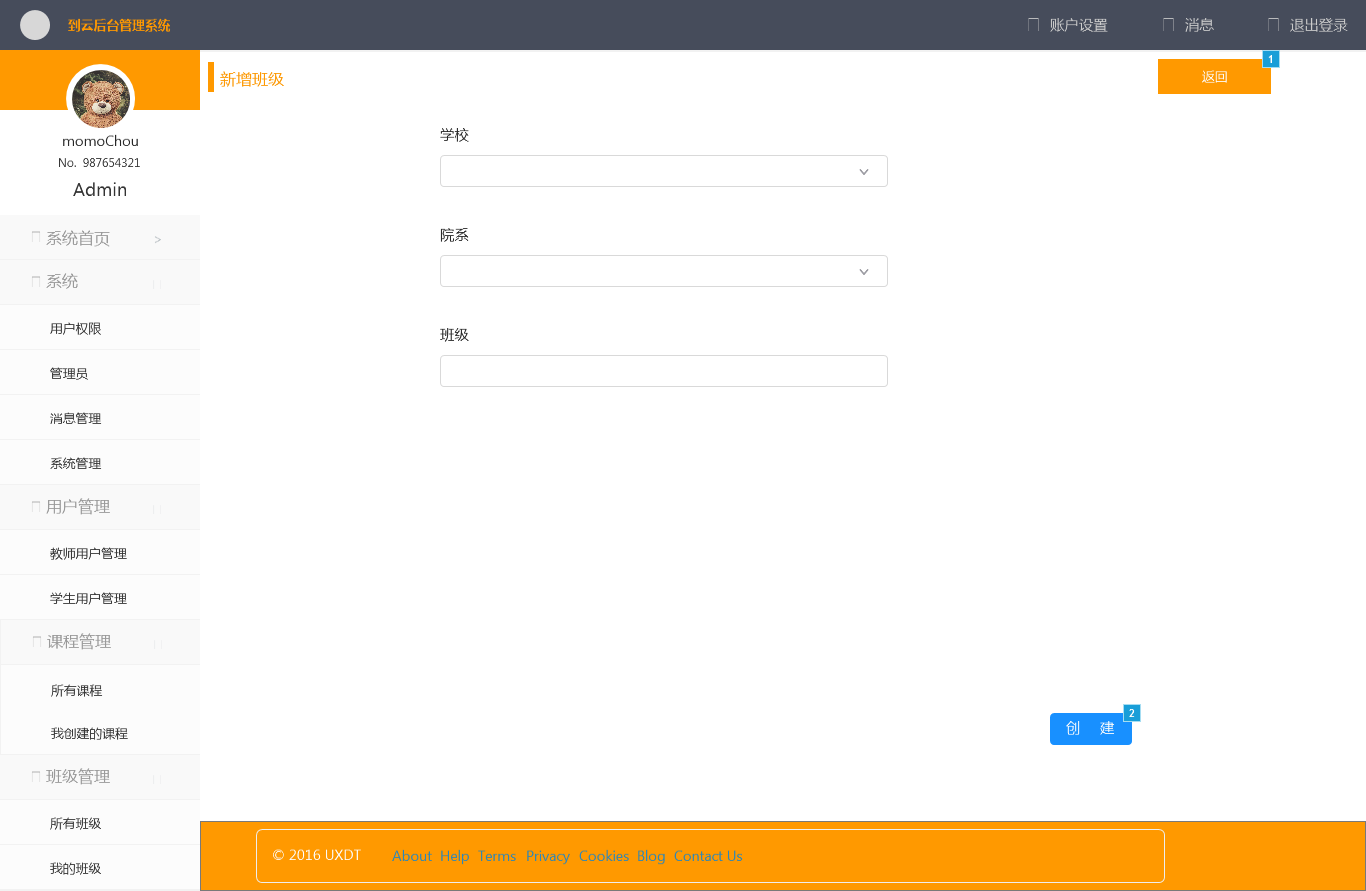
业务流程：无

输入/前置条件：操作用户是教师，且只能对自己上课的班级进行操作

页面逻辑

界面原型：





交互：教师点击每个班级条目中的“查看”，可以查看班级的基本信息，例如学校、院系班级等；点击“删除”可以删除相应的班级；点击“修改”可以修改课程的一些基本信息，例如修改班级名称；点击右上角的“添加”可以创建新的班级，以供学生加入。

字段： 班级信息从数据库中读取；教师若对数据进行操作会被保存到数据库中。

输出/后置条件：

### 学生信息

用户场景：教师从 “班级信息”进入此功能，教师可以对当前某个自己正在授课班级的学生进行管理，增删改查均可。

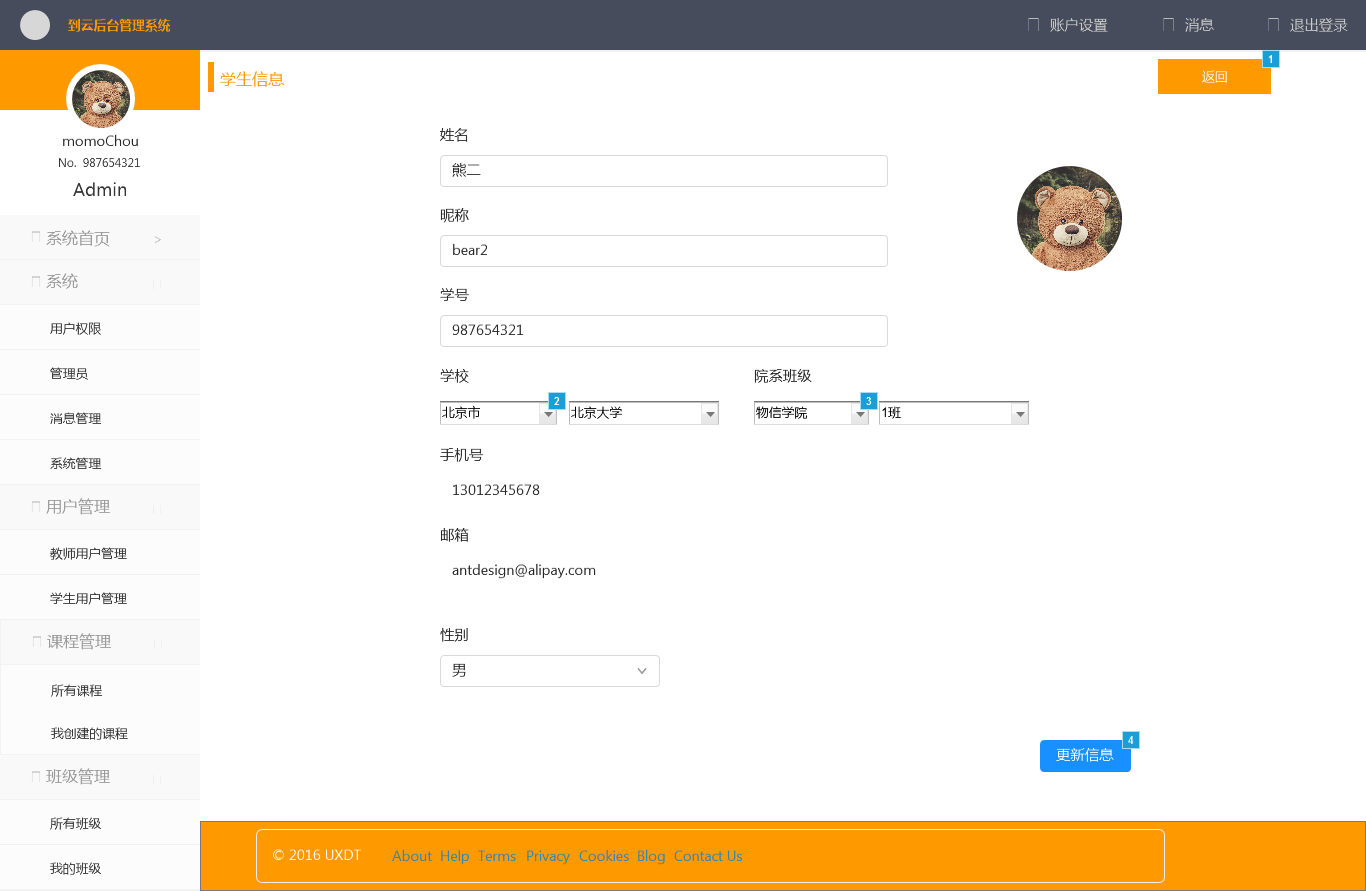
优先级：高

业务流程：无

输入/前置条件：操作用户是教师，且只能对自己上课的班级进行操作

页面逻辑

界面原型：



交互：教师点击每个班级条目中的“学生信息”，可以查看当前班级所有学生的信息。进入“学生信息”页面后，教师点击每个学生条目中的“查看”，可以查看当前学生的信息，例如姓名、学号等；点击“删除”可以删除相应的学生；点击“修改”可以修改课程的一些基本信息，例如学生学号有误时，教师可以修改学生的学号；点击右上角的“添加”可以添加新的学生。

字段： 学生信息从数据库中读取；教师若对数据进行操作会被保存到数据库中。

输出/后置条件：

### 签到情况

用户场景：教师从 “班级信息”进入此功能，教师可以查看当前某个自己正在授课班级的签到情况。

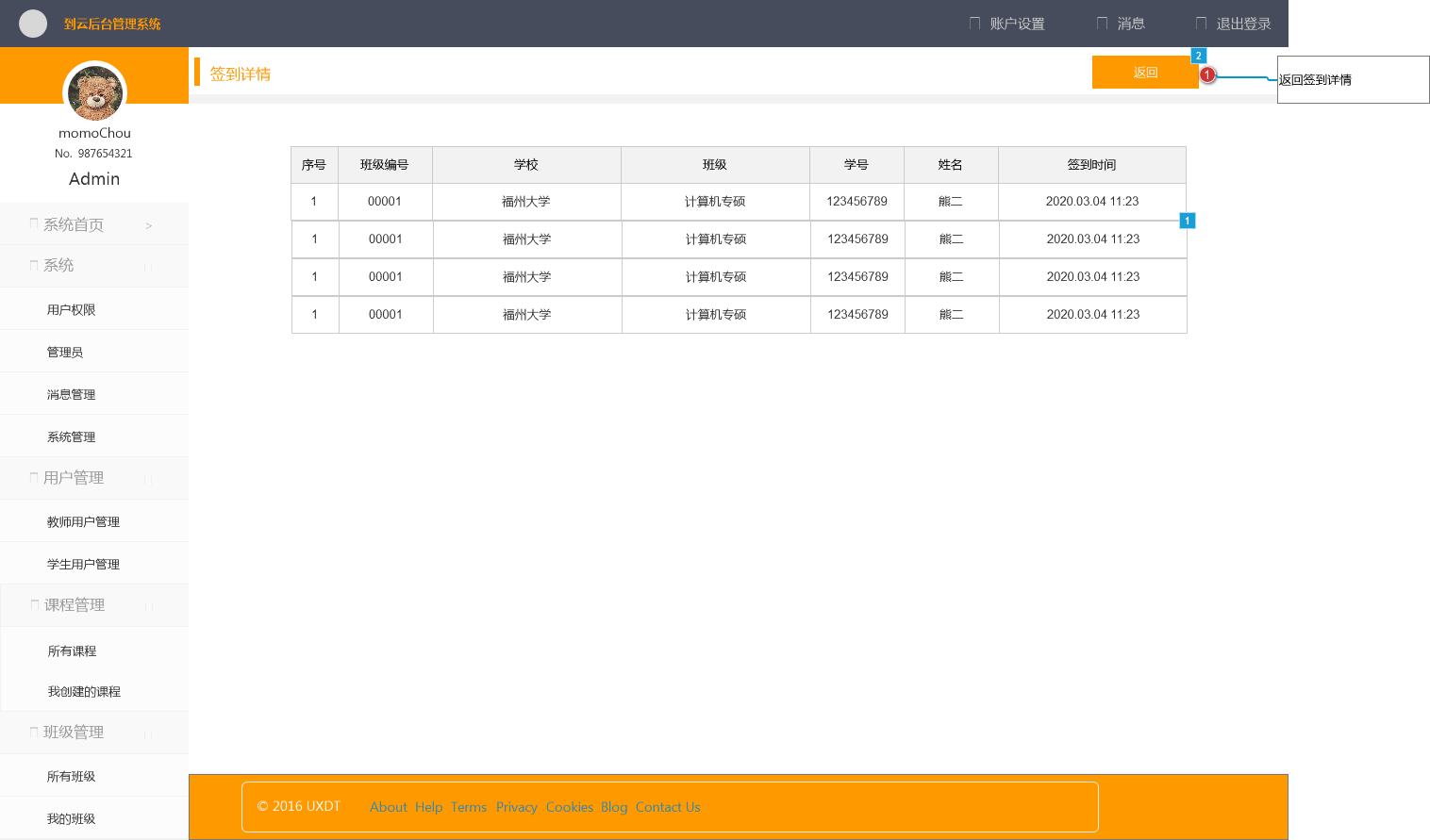
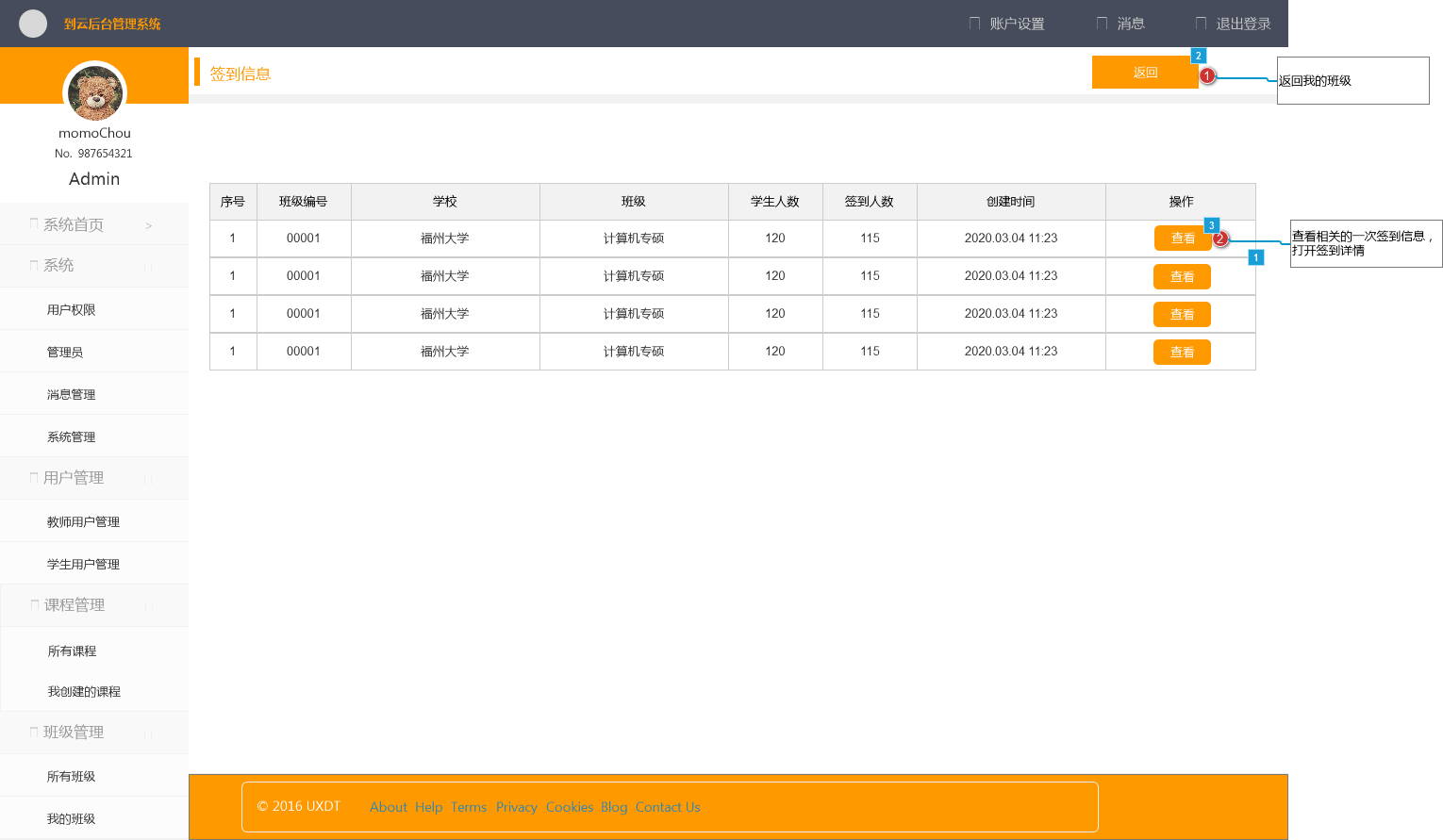
优先级：高

业务流程：无

输入/前置条件：操作用户是教师，且只能对自己上课的班级进行操作

页面逻辑

界面原型：



交互：教师点击每个班级条目中的“签到”，可以查看当前班级所有学生每一次的签到信息。这些信息包括什么时候签到，实到多少人/应到多少人，所有签到了的学生信息

字段： 学生信息从数据库中读取；签到的信息会被保存到数据库中。

输出/后置条件：

# 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。