

大数据学院

软件大型实验周

课程设计说明书

题目： 蓉云课堂

学生姓名： 王天锐 学号： 1607094155

专 业： 物联网工程

指导教师： 于一、柴锐、秦品乐、武瑞娟

2019 年 6 月

目 录

1	设计的主要内容.....	1
1.1	设计目的	1
2	课题实现的软、硬件环境.....	1
2.1	软件环境.....	1
2.1.1	前端.....	1
2.1.2	服务器.....	1
2.2	硬件环境.....	1
3	课题设计	2
3.1	需求分析	2
3.1.1	项目背景	2
3.1.2	业务需求	2
3.1.3	用户需求	2
3.1.4	可行性分析	2
3.1.5	项目概要	3
3.1.6	系统功能结构	3
3.1.7	用户角色和功能摘要	4
3.2	概要设计	5
3.2.1	数据库概念设计	5
3.2.2	数据库逻辑设计	9
3.3	详细设计	13
3.3.1	功能描述	13
3.3.2	系统流程图	14
3.4	编码	15
3.4.1	登录注册模块.....	15
3.4.2	公共模块.....	18
3.4.3	教师功能模块.....	20
3.4.4	学生功能模块	25

软件大型实验周课程设计说明书

4 测试	32
4.1 测试用例设计	32
4.1.1 登陆测试模块.....	32
4.1.2 加入课堂测试模块.....	32
5 心得体会	34
参 考 文 献	35

1 设计的主要内容

1.1 设计目的

本课程设计为学生提供了一个既动手又动脑，独立实践的机会，将课本上的理论知识和实际有机的结合起来，锻炼学生的分析解决实际问题的能力。提高学生适应实际，实践编程的能力。

- 1) 进一步巩固和复习 Android、UML 程序设计的基础知识；
- 2) 培养学生程序设计的方法和能力；
- 3) 了解软件的编制过程及各环节的具体内容；
- 4) 提高学生程序调试技巧、软件设计能力和代码规范化等素质；

提高学生分析问题、解决问题以及实际运用能力。

1.1 设计要求

设计过程：采用设计总体框图，数据库结构和功能结构图。

设计一款基于 Android 平台的云课堂教学助手。

2 课题实现的软、硬件环境

2.1 软件环境

2.1.1 前端

前端整体基于 Android 操作系统实现。主要使用 Java 语言进行逻辑编写、XML 语言进行 UI 编写。使用 Google 公司的 Android Studio 编译器进行编码。最佳适配 API 版本为 27、Gradle 版本为 4.6。

2.1.2 服务器

后台服务器基于 DjangoWeb 框架以及 MySQL 数据库设计与开发。主要使用 Python 语言进行逻辑编写、SQL 语言进行数据库操作。使用 JetBrains 公司的 Pycharm 编译器进行编码。编码完成后移植到阿里云应用服务器进行部署。

2.2 硬件环境

Android 前端使用 XiaoMi 6x 进行测试、编码使用常规笔记本电脑。

3 课题设计

3.1 需求分析

3.1.1 项目背景

手机目前已经成为了当代人的必需品，人们已经越来越习惯通过手机解决日常生活需求。但是目前大多高校对课堂上学生手机的使用情况管理欠佳，学生们无法合理地发挥手机便利的功能于学习之上。一方面，老师们对于学生手机的管理也非常艰难，学生多、手机多、代课班级多的问题也让诸多老师颇为头疼。另一方面，学生上课用手机也不一定全是玩游戏、刷朋友圈，也有查阅资料、记录笔记、拍照留题等用途，若上课没有手机也会带来诸多不便。

目前很多高校老师抱着宁为玉碎不为瓦全的心态让同学们上课期间上交手机，这样虽能解决学生上课玩游戏、聊天的情况。但是这样做一方面舍弃了手机便携的查询资料功能和拍照记录功能，另一方面老师们不能够有效地、针对性地知道哪些同学拿到手机在课堂上是有帮助的哪些是有坏处的。一款能和合理统计学生上课手机使用情况、方便学生查阅资料、方便学生提出问题、提交作业，的云班课 APP 成为了广大高校的迫切需求。

3.1.2 业务需求

时间需求：在网络畅通的前提下输入、输出相应时间不超过 1.5 秒。

并发性需求：能多人同时操作并且不卡顿。

3.1.3 用户需求

UI 界面美观大方、追求简洁；操作易理解、人性化；App 操作方便、不卡顿。

3.1.4 可行性分析

该云课堂系统是基于 Android、Python、DJango 和 MySQL 等技术进行设计与开发的，整个云课堂系统分为客户端和服务端。客户端使用 Android 来实现，服务端用 Python 进行开发。

Android 系统是基于 Linux 的，主要支持手机而且是开放源代码的操作系统。整个系统由下到上分为 Linux 内核、Libraries(库) + Android Runtime(Android 运行时)、APPLication Framework(应用程序框架层)、APPLication(应用程序层)，我们的程序属于应用程序层开发。

服务端采用 Python 语言进行开发，它是一种通用脚本语言，语法简单但是功能强大，便于快速入门。服务器框架采用 Django，Python 下有许多款不同的 Web 框架。Django 是重量级选手中最有代表性的一位。Django 是一个开放源代码的 Web 应用框架，遵守 BSD 版权，采用了 MVC 的软件设计模式。

数据库该系统使用 Mysql。Mysql 是一个关系型数据库管理系统，其体积小、速度快、总体拥有成本低。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。操作简单、便于使用。

3.1.5 项目概要

《蓉云课堂》是在 Android 操作系统上运行的一款云课堂助学软件。针对目前高校上课同学管理以及课堂上手机管理的问题，本系统提供了相应的解决方法。该系统主要用户分为学生和老师两类。针对课堂管理不方便的问题，本软件为老师用户提供创建云课堂、上传资料、发布通知、发起签到、发布作业、作业评分等功能。相对应地，学生用户可以通过输入邀请码加入课堂、下载资料、查看通知、签到、上传作业、在师生交流区发表意见。针对学生上课手机管理问题，本软件为老师们提供学生上课手机电量使用情况统计、自动根据学生分数安排学生座位的功能。

3.1.6 系统功能结构

本系统包含 Android 移动端和服务端，其中移动端分为教师与学生两大模块。系统整体结构如图 3.1 所示。

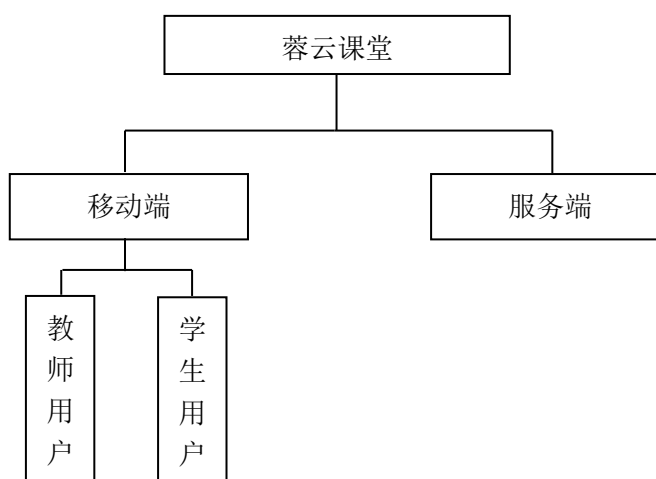


图 3.1 系统整体结构图

该系统主要包含课堂创建与加入、课程表、课堂资料发表、课堂交流、作业、通知、电量记录、打分、自动排座、通讯录等功能。整体功能结构图如图 3.2 所示。

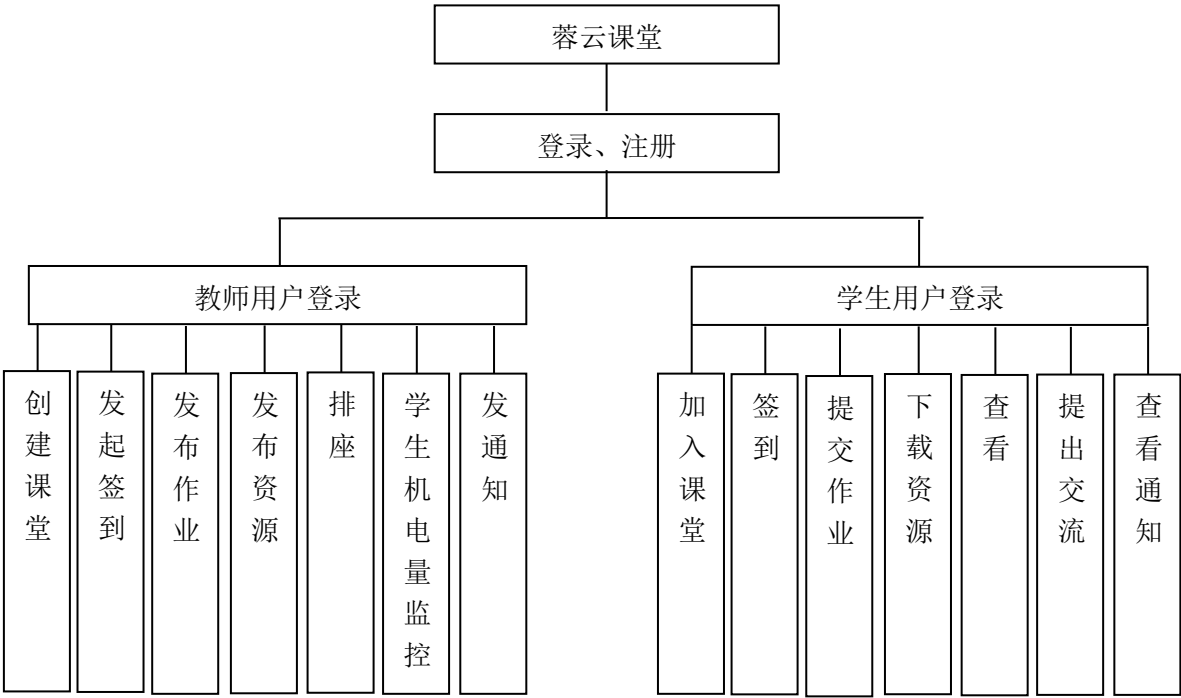


图 3.2 整体功能结构图

3.1.7 用户角色和功能摘要

用户角色说明如表 3.1 所示。

表 3.1 用户角色说明

用户角色	用户描述
教师用户	该类用户拥有手机端的所有操作权限，便于老师管理同学。
学生用户	该类用户只拥有下载资源、提交作业等必要操作权限。

功能摘要说明如表 3.2 所示。

表 3.2 功能摘要说明

功能模块	功能描述	优先级
创建与加入课堂	创建后会获取到班课邀请码，以供学生加入课堂。学生通过邀请码加入课堂。	高
签到	教师可以在自己的班课上发布签到，发布成功后会获得签到码，供学生们签到。	高
作业功能	教师可以在班课里发布作业，作业包括题目和内容。可就学生提交的作业回答进行打分。	中
资料管理	教师可以上传资料到班课里。学生可下载查看。	高

通知	教师可以发布通知	高
信息查看	用户可以查看班级其他同学信息。	高
课表	用户可以自定义课程表	中
电量可视化	老师端可以查看同学们手机电量的走势以及手机充电状态。	高
自动排座	可以一键根据学生学习情况为学生安排座位	中
讨论交流	师生可以在交流区发帖交流，老师可以在其中为学生 的发言打分。	中

3.2 概要设计

3.2.1 数据库概念设计

本系统中包含 14 个实体：用户实体、课程实体、通知实体、作业实体、回答实体、交流主题实体、交流消息实体、发起签到实体、学生签到实体、分数实体、资源实体、课表实体、电量实体、座位实体，根据需求分析的结果以上六个实体所对应的 ER 属性图如 3.3、3.4、3.5 所示。

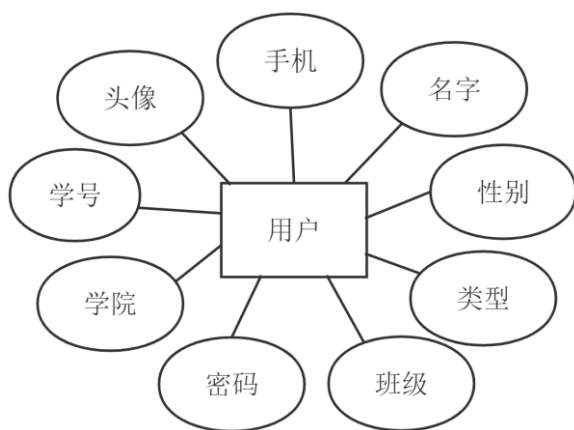


图 3.3 用户实体属性图

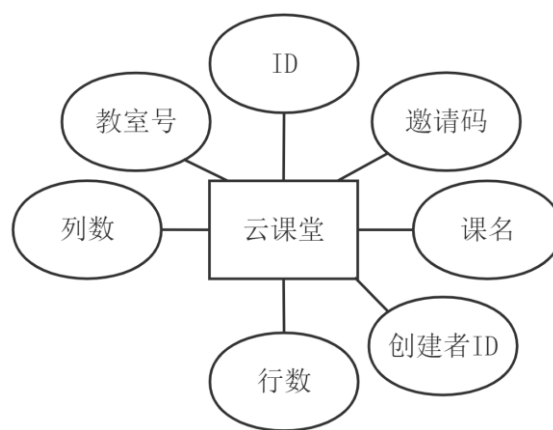


图 3.4 云课堂实体属性图

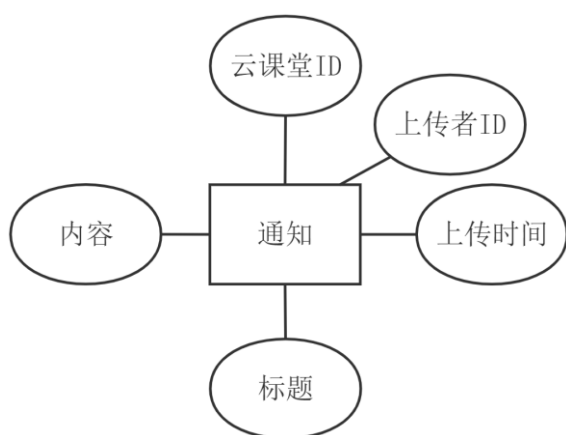


图 3.5 通知实体属性图

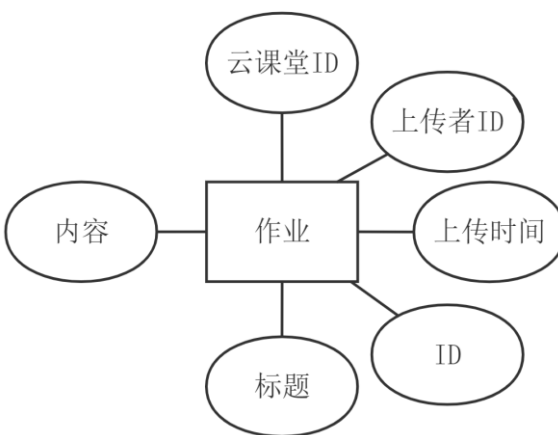


图 3.6 作业实体属性图

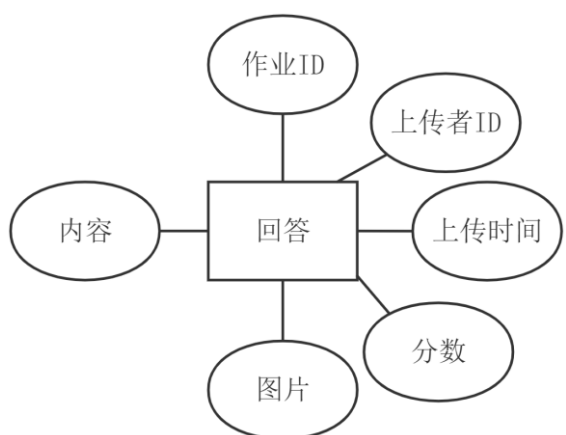


图 3.7 回答实体属性图

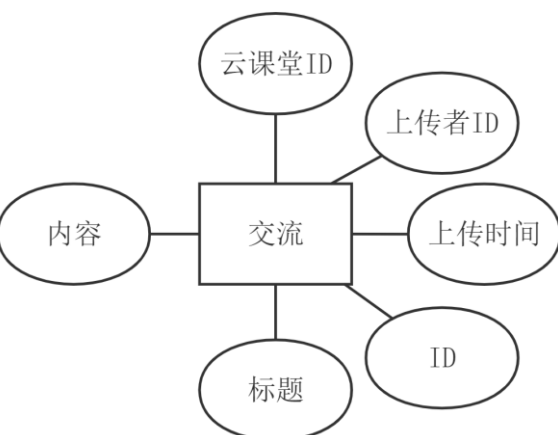


图 3.8 交流实体属性图

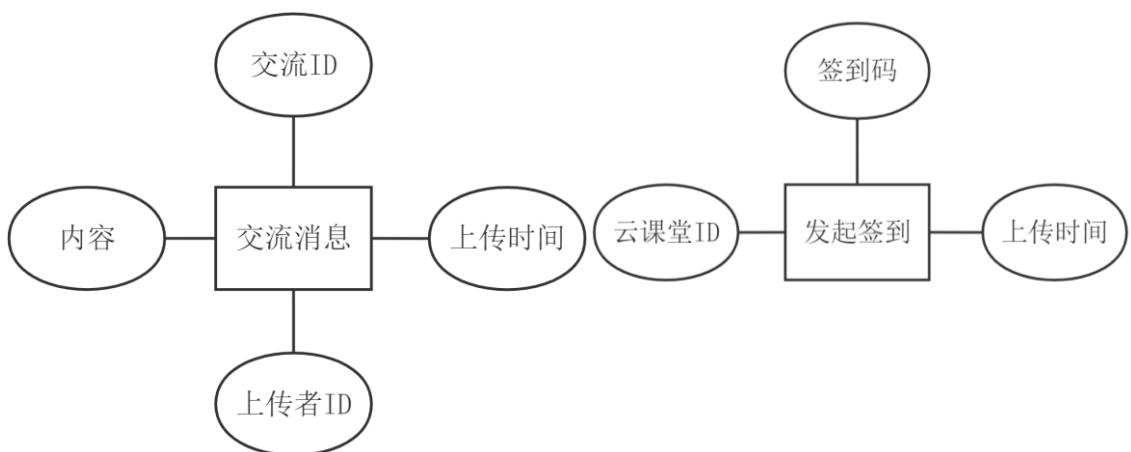


图 3.9 交流消息实体属性图

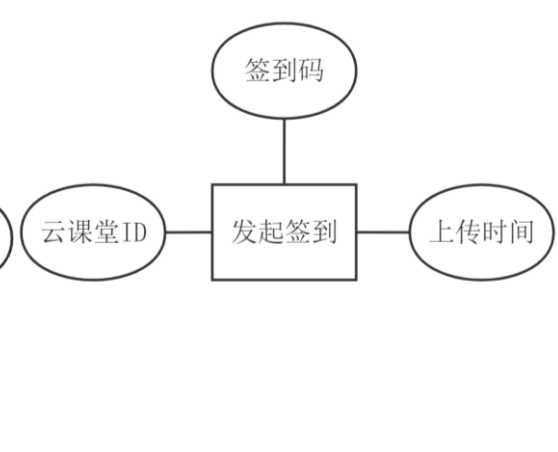


图 3.10 发起签到实体属性图

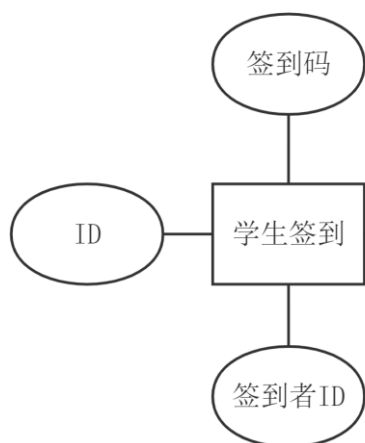


图 3.11 学生签到实体属性图

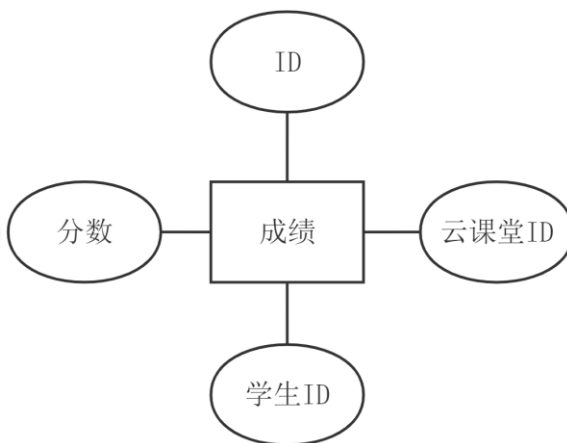


图 3.12 成绩实体属性图

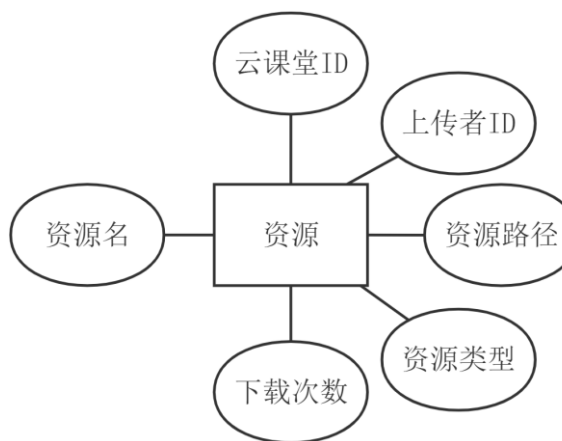


图 3.13 资源实体属性图

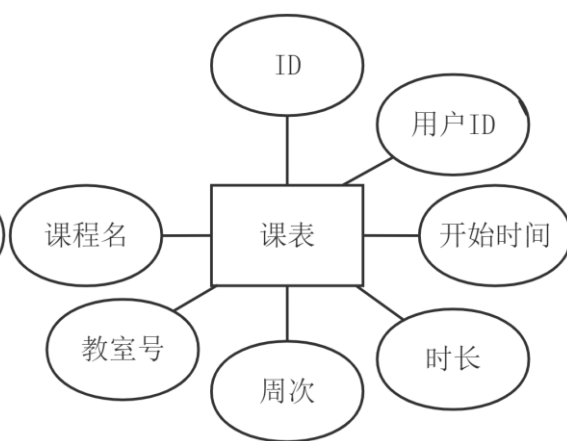


图 3.14 课表实体属性图

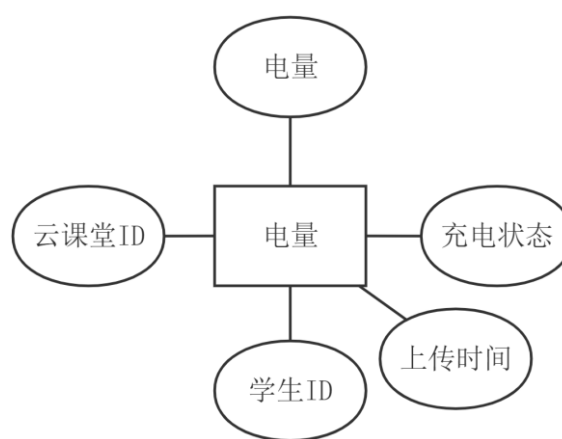


图 3.15 电量实体属性图

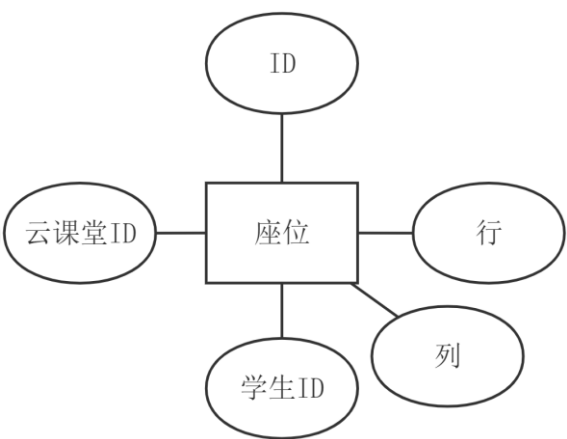


图 3.16 座位实体属性图

系统整体 E-R 图如图 3.17 所示。

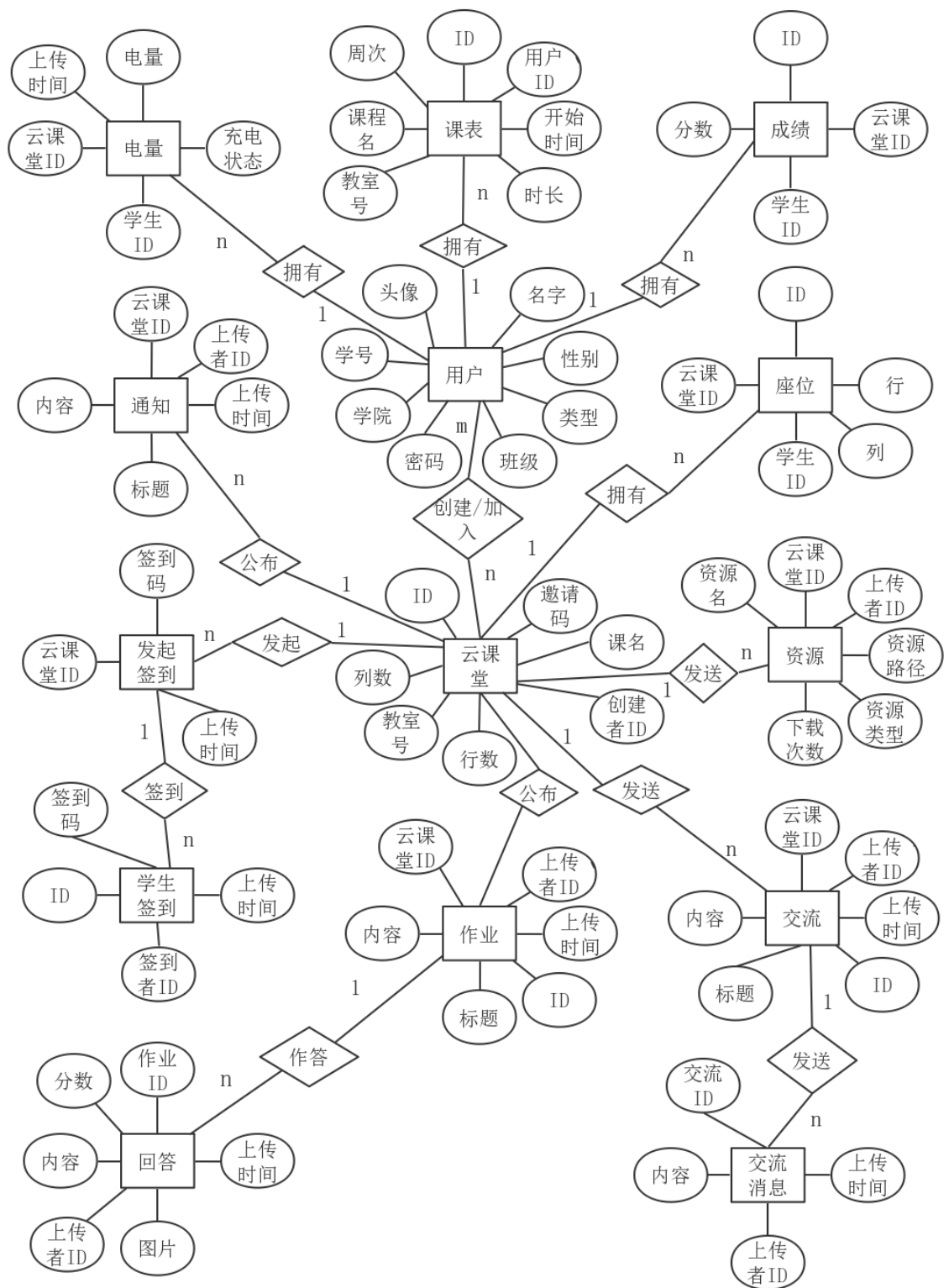


图 3.17 系统整体 E-R 图

3.2.2 数据库逻辑设计

根据 E-R 图向关系模型的转换原则，一个实体型转换为一个关系模式，实体的属性就是关系的属性。因此按照图 3.18 中的整体 E-R 图，本数据库系统应包括 15 张表：用户、课程表、电量、成绩、通知、座位、发起签到、学生签到、云课堂、资源、学生加入课堂、作业、交流、交流信息、回答。

表 3.3 用户(User)

列名	数据类型	说明	约束
phone	Char (20)	手机号	主码
name	Char (20)	名字	非空
sex	Char (5)	性别	非空
type	Int	类型	非空
classNumber	Char (20)	班级号	非空
password	Char (20)	密码	非空
academy	Char (20)	学院	非空

表 3.4 云班课(Course)

列名	数据类型	说明	约束
ID	Char (20)	课程 ID	主码
inviteCode	Char (6)	邀请码	非空
name	Char (20)	课程名	非空
teacherName	Char (20)	教师名	非空
row	Int	行数	非空
col	Int	列数	非空
classroom	Char (20)	教室	非空

表 3.5 学生上课(Student2Course)

列名	数据类型	说明	约束
ID	Char (20)	编码	主码
StudentID	Char (20)	学生主码	外键
CourseID	Char (20)	课程主码	外键

表 3.6 通知 (Inform)

列名	数据类型	说明	约束
ID	Char (20)	编码	主码
TeacherID	Char (20)	老师主码	外键
Time	DateTime	上传时间	非空
Title	Char (20)	标题	非空
Content	Char (20)	内容	非空
CourseID	Char (20)	课程主码	外键

表 3.7 作业 (HomeWork)

列名	数据类型	说明	约束
ID	Char (20)	编码	主码
TeacherID	Char (20)	老师主码	外键
Time	DateTime	上传时间	非空
Title	Char (20)	标题	非空
Content	Char (20)	内容	非空
CourseID	Char (20)	课程主码	外键

表 3.8 回答 (Answer)

列名	数据类型	说明	约束
ID	Char (20)	编码	主码
TeacherID	Char (20)	老师主码	外键
Time	DateTime	上传时间	非空
Content	Char (20)	内容	非空
HomeWorkID	Char (20)	作业主码	外键

表 3.9 交流 (Communication)

列名	数据类型	说明	约束
ID	Char (20)	编码	主码
UserID	Char (20)	用户主码	外键
Time	DateTime	上传时间	非空

Content	Char (20)	内容	非空
CourseID	Char (20)	课程主码	外键

表 3.10 交流内容 (CommunicationItem)

列名	数据类型	说明	约束
ID	Char (20)	编码	主码
UserID	Char (20)	用户主码	外键
Time	DateTime	发起时间	非空
Content	Char (20)	内容	非空
CommunicationID	Char (20)	交流主码	外键

表 3.11 发起签到 (Sign)

列名	数据类型	说明	约束
SignCode	Char (20)	签到码	主码
Time	DateTime	发起时间	非空
CourseID	Char (20)	课程主码	外键

表 3.12 学生签到 (StudentSign)

列名	数据类型	说明	约束
SignCode	Char (20)	签到码	外键
Time	DateTime	签到时间	非空
StudentId	Char (20)	学生主码	外键

表 3.13 资源 (MySource)

列名	数据类型	说明	约束
ID	Char (20)	编码	主码
UserID	Char (20)	上传者主码	外键
Time	DateTime	上传时间	非空
Path	Char (100)	资源下载 URL	非空
Name	Char (20)	资源名	非空
CourseID	Char (20)	课程主码	外键

3.14 电量 (Power)

列名	数据类型	说明	约束
ID	Char (20)	编码	主码
StudentID	Char (20)	学生主码	外键
Time	DateTime	上传时间	非空
Level	Int	电量	非空
Status	Char (20)	状态	非空
CourseID	Char (20)	课程主码	外键

3.15 座位 (Seat)

列名	数据类型	说明	约束
ID	Char (20)	编码	主码
StudentID	Char (20)	学生主码	外键
Row	Int	行数	非空
Col	Int	列数	非空
CourseID	Char (20)	课程主码	外键

表 3.16 课表 (CourseModel)

列名	数据类型	说明	约束
ID	Char (20)	编码	主码
Week	Int	周次	非空
Section	Int	开始时间	非空
CourseName	Char (20)	课程名	非空
classroom	Char (20)	教室号	非空

表 3.17 成绩 (Score)

列名	数据类型	说明	约束
ID	Char (20)	编码	主码
StudentID	Char (20)	学生主码	外键
Score	Float	成绩	非空
CourseID	Char (20)	课程主码	外键

3.3 详细设计

3.3.1 功能描述

蓉云课堂具体包含如下功能：

- 1) 教师可以创建云班课，创建后会获取到班课邀请码，以供学生加入课堂。创建云课堂的时候需要输入教室座位行数和排数，以便自动为学生排座位。（学生不可创建课堂）。
- 2) 教师可以在自己的班课上发布签到，发布成功后会获得签到码，以供学生们签到。
- 3) 教师可以在班课里发布通知。通知包括标题和内容。
- 4) 教师可以在班课里发布作业，作业包括题目和内容。
- 5) 教师可以就学生提交的作业回答进行打分。
- 6) 教师可以上传资料到班课里。
- 7) 教师可以查看班课同学名单和各位同学的成绩以及联系方式等详细信息。
- 8) 教师可以一键自动为学生安排座位，座位根据学生的成绩和课堂表现进行安排，实现差优均匀分配。
- 9) 学生在界面点击按钮输入邀请码加入老师创建的云课堂。
- 10) 学生可以通过签到码进行课堂签到。
- 11) 学生可以查看老师发布的通知。
- 12) 学生可以查看老师布置的作业，并且在相应题目下提交自己的答案，答案分为内容和图片。
- 13) 学生可以查看自己以及班级其他同学的成绩。
- 14) 学生可以下载老师上传的资料，并进行预览。
- 15) 学生每节课上课前可以看到自己应该坐的座位。
- 16) 所有用户可以登录注册。登录注册使用手机号作为账号，注册时需要设置用户的类型（教师/学生）。

17) 所有用户可以在交流答疑区开帖子，其余人可以在其中进行交流，老师可以为学生的单个帖子进行打分。

18) 所有用户可以制作属于自己的课表。

3.3.2 系统流程图

系统整体流程图如图 3.18 所示。

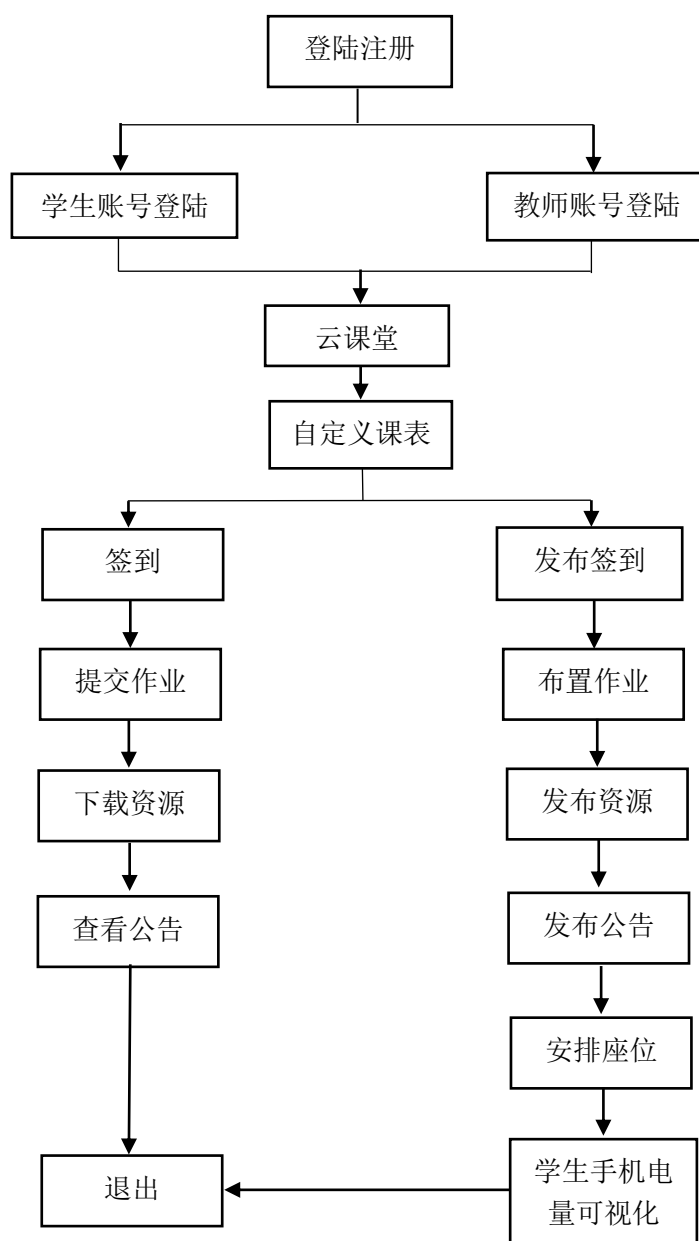


图 3.18 系统流程图

3.4 编码

3.4.1 登录注册模块

(1) 功能描述

该模块主要实现用户的登录和注册功能，系统需要注册账号，用户使用该系统必须登录，登录时向服务器请求数据是否存在，密码是否匹配，若匹配，返回登录信息。

(2) 流程图

登录注册模块流程图如图 3.19 所示。

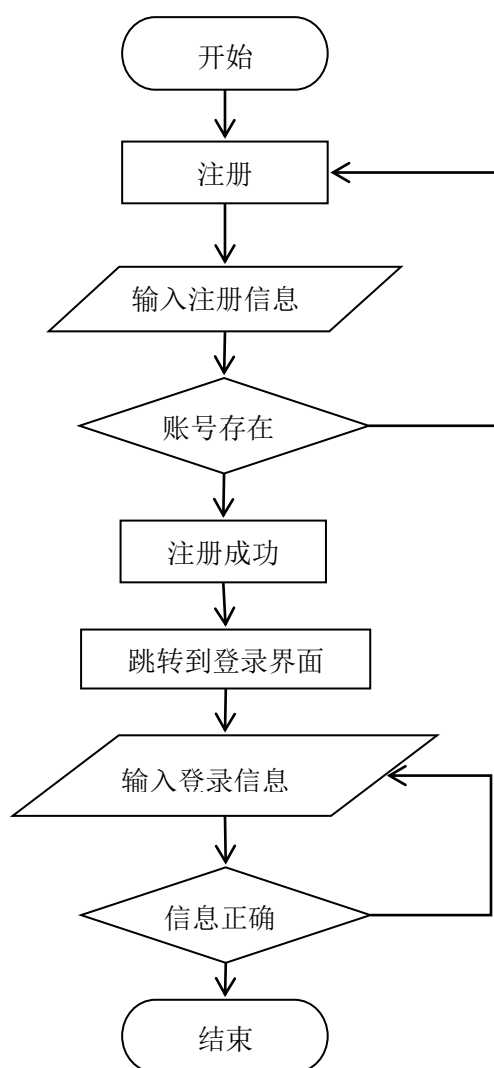


图 3.19 登录注册模块流程图

(3) 界面设计

登录和注册界面设计如图 3. 20, 3. 21 所示。

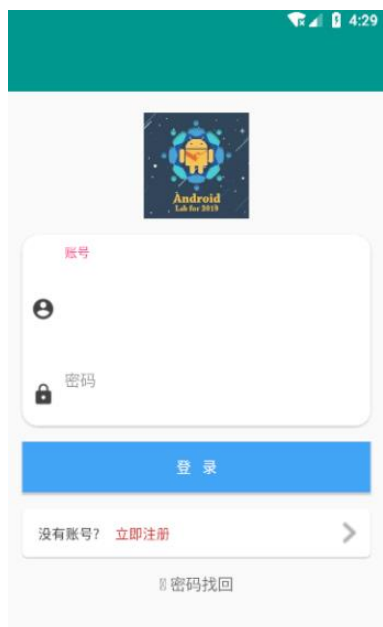


图 3. 20 登录模块界面图



图 3. 21 注册模块界面图

Web 端登录、注册界面设计如图 3. 22, 3. 23 所示。



图 3. 22 Web 端登录界面



图 3. 23 Web 端注册界面

(4) 主要代码实现

```
String phone = loginEdtUsername.getText().toString()
    .trim();

String password = loginEdtPassword.getText().toString()
    .trim();
```

```
HashMap<String, String> data = new HashMap<>();
data.put("phone", phone);
data.put("passWord", password);
WebUtils.login(data, new Callback() {
    @Override
    public void onFailure(Call call, IOException e) {
    }
    @Override
    public void onResponse(Call call, Response response)
        throws IOException {
        Gson gson = new Gson();
        JsonObject jsonObject = getJsonObject(response);
        User user = gson.fromJson(jsonObject
            .get("values"), User.class);
        CloudClass.user = user;
        handler.sendMessage(jsonObject.get("result").getAsInt());
    }
});
public static void login(HashMap<String, String> data, Callback callback)
{
    Request request = new Request.Builder().post(getBody(data))
        .url(ConstantUtils.webHost + "login/").build();
}
public static void register(HashMap<String, String> data, Callback
callback) {
    Request request = new Request.Builder().post(getBody(data))
        .url(ConstantUtils.webHost + "register/").build();
    call.enqueue(callback);
}
```

3.4.2 公共模块

(1) 功能描述

老师与学生均可在 APP 里参看自己加入了的云班课、点击进入、自定义课程表、查看个人信息等。

(2) 流程图

公共模块流程图如图 3.24 所示。

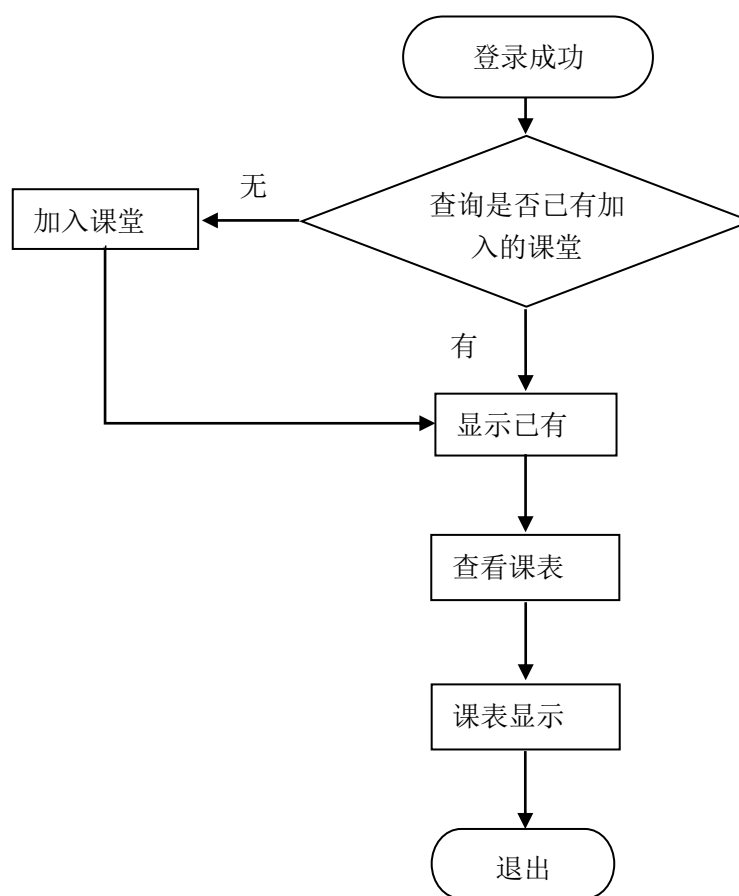


图 3.24 公共模块流程图

(3) 界面设计

课堂列表、创建课表、课表界面和个人信息界面如图 3.25, 3.26, 3.27, 3.28 所示。



图 3.25 课程列表界面



图 3.26 添加课表界面

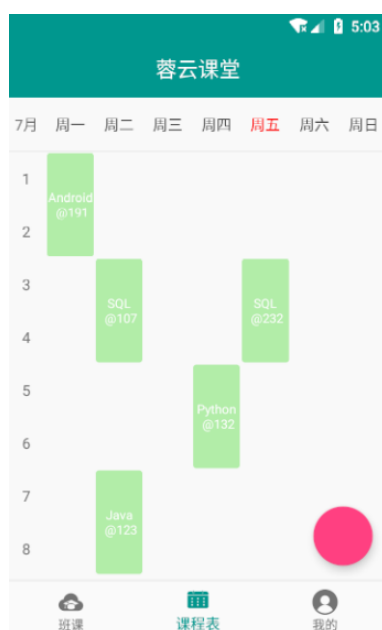


图 3.27 课程表界面



图 3.28 个人信息界面

(4) 主要代码实现

```
public static void addCourseTable(HashMap<String, String> data, Callback
callback) {
```

```
    Request request = new Request.Builder().post(getBody(data))
        .url(ConstantUtils.webHost + "add_courseTable/").build();
```

```
Call call = mClient.newCall(request);
call.enqueue(callback);
}

public static void getCourseTable(HashMap<String, String> data, Callback
callback) {
    Request request = new Request.Builder().post(getBody(data))
        .url(ConstantUtils.webHost + "get_courseTable/").build();
    Call call = mClient.newCall(request);
    call.enqueue(callback);
}

public static void getUser(HashMap<String, String> data, Callback callback)
{
    Request request = new Request.Builder().post(getBody(data))
        .url(ConstantUtils.webHost + "get_user/").build();
    Call call = mClient.newCall(request);
    call.enqueue(callback);
}

public static void getCourseList(HashMap<String, String> data, Callback
callback) {
    Request request = new Request.Builder().post(getBody(data))
        .url(ConstantUtils.webHost + "get_course/").build();
    Call call = mClient.newCall(request);
    call.enqueue(callback);
}
```

3.4.3 教师功能模块

(1) 功能描述

教师用户可以发起签到、发布公告、发布作业、打分、发起交流、上传资源、查看学生上课手机情况。

(2) 流程图

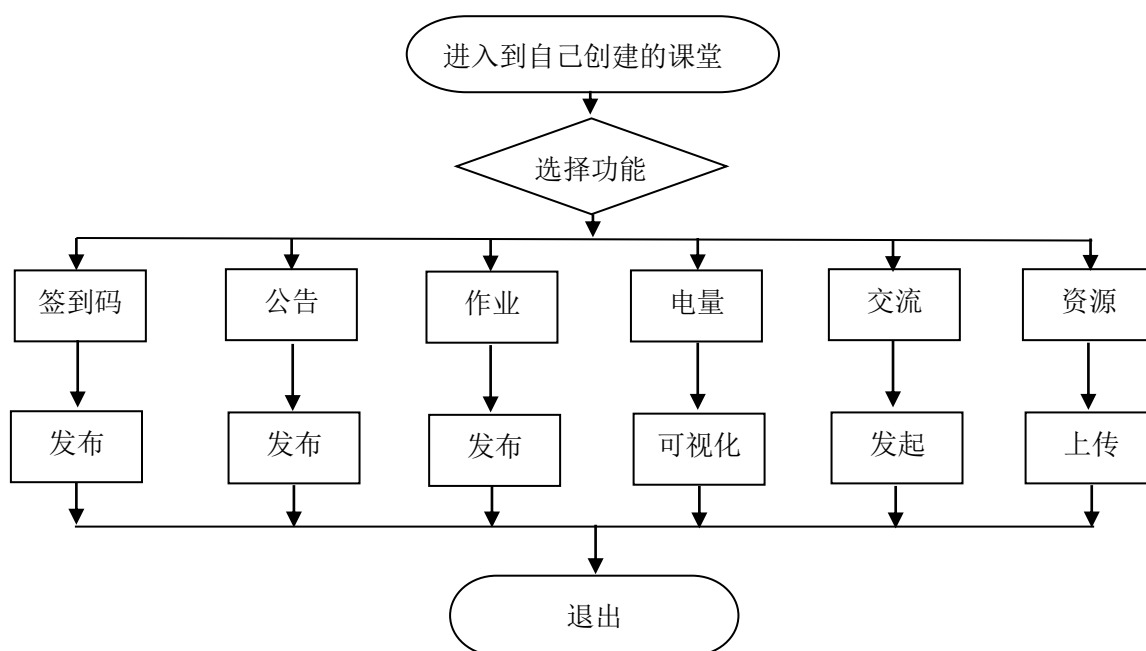


图 3.29 教师功能流程图

(3) 界面设计

发起签到、发布公告、发布作业、打分、发起交流、上传资源、查看学生上课手机情况界面分别为图 3.30, 3.31, 3.32, 3.33, 3.34, 3.35, 3.36, 3.37 所示。



图 3.30 发起签到



图 3.31 发布公告



图 3.32 公布作业



图 3.33 打分



图 3.34 提出交流话题



图 3.35 交流详细



图 3.36 上传资源



图 3.37 电量可视化

(4) 主要代码实现

```
public static void addPower(HashMap<String, String> data, Callback
callback) {
    Request request = new Request.Builder().post(getBody(data))
        .url(ConstantUtils.webHost + "addPower/").build();
    Call call = mClient.newCall(request);
    call.enqueue(callback);
}

public static void getPower(HashMap<String, String> data, Callback
callback) {
    Request request = new Request.Builder().post(getBody(data))
        .url(ConstantUtils.webHost + "getPower/").build();
    Call call = mClient.newCall(request);
    call.enqueue(callback);
}

public static void upSign(HashMap<String, String> data, Callback callback)
{

```

```
Request request = new Request.Builder().post(getBody(data))
    .url(ConstantUtils.webHost + "upSign/").build();
Call call = mClient.newCall(request);
call.enqueue(callback);
}

public static void upInform(HashMap<String, String> data, Callback
callback) {
    Request request = new Request.Builder().post(getBody(data))
        .url(ConstantUtils.webHost + "create_inform/").build();
    Call call = mClient.newCall(request);
    call.enqueue(callback);
}

public static void addAnswer(String homeworkId, String uperId, String
content, String path, File source, Callback callback) {
    MediaType mediaType = MediaType.parse("multipart/form-data;
boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW");

    RequestBody requestBody = new MultipartBody.Builder()
        .setType(MultipartBody.FORM)
        .addFormDataPart("homework_id", homeworkId)
        .addFormDataPart("content", content)
        .addFormDataPart("userId", uperId)
        .addFormDataPart("image",
            RequestBody.create(mediaType, source)
        )
        .build();
    Call call = mClient.newCall(request);
    call.enqueue(callback);
}
```

3.4.4 学生功能模块

(1) 功能描述

学生用户可以进行签到、查看公告、提交作业、交流、下载资源、查看座位。

(2) 流程图

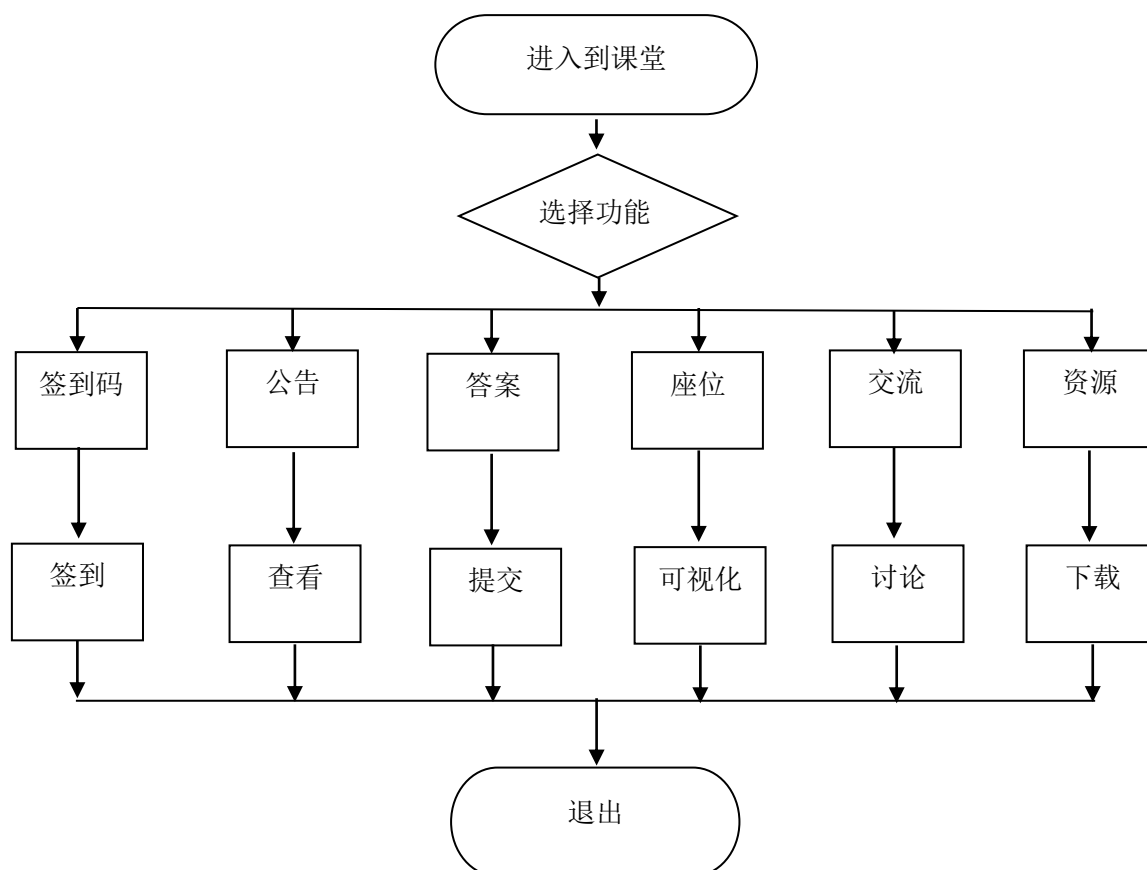


图 3.38 学生模块流程图

(3) 界面设计

学生用户进行签到、查看公告、提交作业、交流、下载资源、查看座位界面如图 3.39，3.40，3.41，3.42，3.43，3.44 所示。



图 3.39 签到界面



图 3.40 查看公告界面



图 3.41 提交作业界面



图 3.42 交流界面



图 3.43 下载资源界面



图 3.44 查看座位界面

(4) 主要代码实现

```
public static void studentSign(HashMap<String, String> data, Callback
callback) {
    Request request = new Request.Builder().post(getBody(data))
        .url(ConstantUtils.webHost + "studentSign/").build();
    Call call = mClient.newCall(request);
    call.enqueue(callback);
}

public static void getInform(HashMap<String, String> data, Callback
callback) {
    Request request = new Request.Builder().post(getBody(data))
        .url(ConstantUtils.webHost + "get_inform/").build();
    Call call = mClient.newCall(request);
    call.enqueue(callback);
}

public static void sendCommunicationItem(HashMap<String, String> data,
Callback callback) {
    Request request = new Request.Builder().post(getBody(data))
```

```
        .url(ConstantUtils.webHost
"create_communicationitem/").build();
        Call call = mClient.newCall(request);
        call.enqueue(callback);
    }

    public static void addAnswer(String homeworkId, String uperId, String
content, String path,

                                File source, Callback callback) {

        MediaType mediaType = MediaType.parse("multipart/form-data;
boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW");

        RequestBody requestBody = new MultipartBody.Builder()

            .setType(MultipartBody.FORM)

            .addFormDataPart("homework_id", homeworkId)

            .addFormDataPart("content", content)

            .addFormDataPart("userId", uperId)

            .addFormDataPart("image",

                RequestBody.create(mediaType, source)

            )

            .build();

        Call call = mClient.newCall(request);
        call.enqueue(callback);
    }

    public static void getSeat(HashMap<String, String> data, Callback callback)
    {

        Request request = new Request.Builder().post(getBody(data))

            .url(ConstantUtils.webHost + "getSeat/").build();

        Call call = mClient.newCall(request);
        call.enqueue(callback);
    }
}
```


3.4.5 服务端模块

(1) 功能描述

通过前端对后台数据进行管理。

(2) 界面设计

数据库表展示、增加、删除界面如图 3.45，3.46，3.47 所示



Answers	修改
Communicaiton items	修改
Communications	修改
Course models	修改
Courses	修改
Home works	修改
Informs	修改
My sources	修改
Powers	修改
Scores	修改
Seats	修改
Signs	修改
Student signs	修改
Student2 courses	修改
Users	修改

图 3.45 后台数据查询



图 3.46 增添数据

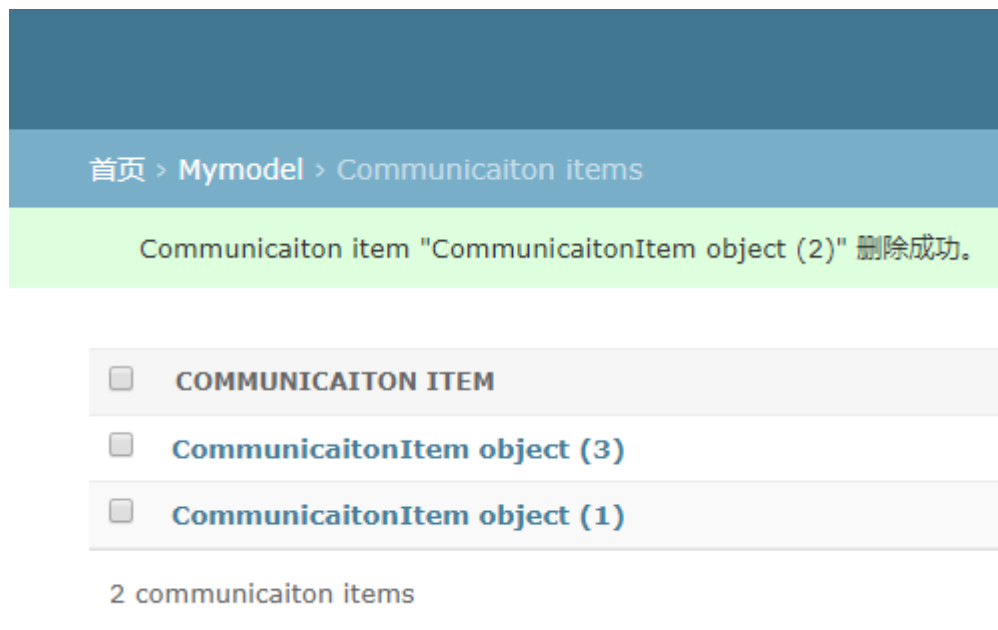


图 3.47 删除数据

(3) 代码实现

```
<div id="content" class="colMS">
  <div id="content-main">
    <div class="app-MyModel module ">
      <table>
```

```
{% for item in values %}

    <tr>

        <td>{{ item }}</td>

    </tr>

{% endfor %}

</table>

</div>

</div>

</div>

var containerTable = document.getElementById("container_table");
var tbody = document.createElement("tbody");
for (const name in nameList) {
    if (nameList.hasOwnProperty(name)) {
        const element = nameList[name];
        var tr = document.createElement("tr");
        var th = document.createElement("th");
        tr.appendChild(th);
        var a_name = document.createElement("a")
        var td_add = document.createElement("td");
        var a_add = document.createElement("a")
        var td_edit = document.createElement("td");
        td_edit.setAttribute("class", "changelink");
        var a_edit = document.createElement("a")
        tr.appendChild(td_add);
        tr.appendChild(td_edit);
        tr.setAttribute("class", "model-student2course");
        tbody.appendChild(tr);
    }
}
```

4 测试

4.1 测试用例设计

4.1.1 登陆测试模块

用户名和密码具有这几种情况：用户名是否存在、密码是否匹配。对以上各种情况进行组合，观察不同的操作的响应是否和预期的一样，测试用例如表 4.1 所示。

表 4.1 登录测试用例表

用户名	密码	预期结果	实际结果
15008461730	123456	匹配成功	匹配成功
15008461730	111111	密码错误	密码错误
12345	111111	用户名不存在	用户名不存在
123	123456	密码错误	密码错误

登陆测试结果如图 4.1、4.2 所示。

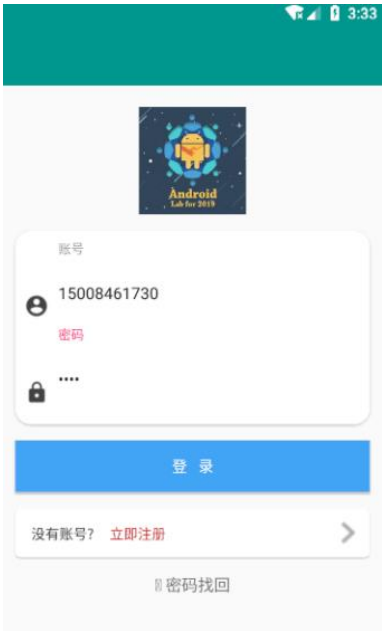


图 4.1 输入错误的密码



图 4.2 提示密码错误

4.1.2 加入课堂测试模块

加入课堂需要学生输入邀请码，其具有两种情况：邀请码不存在、邀请码正确。对以上各种情况进行组合，观察不同的操作的响应是否和预期的一样，测试用例如表 4.2 所示。

表 4.2 加入课堂测试用例表

邀请码	预期结果	实际结果
16979	加入课堂成功	加入课堂成功
12345	邀请码不存在	邀请码不存在

登陆测试结果如图 4.3、4.4 所示。



图 4.3 输入错误的邀请码



图 4.4 提示邀请码错误

5 心得体会

本次 Android 课程设计通过设计一个云课堂 APP 来满足多种需求以供使用。在本次课程设计中也遇到了很多问题,如数据库连接不上,服务器与前端交互问题等。这次课设证实提供了一个很好的机会。通过这次的系统设计,在很多方面都有所提高。综合运用所学知识的理论知识实际训练从而培养和提高独立工作的能力,巩固所学的知识,掌握系统程序的编排和运行,使自己的独立思考能力有了显著提高。从各种文档的阅读到开始的需求分析、概念结构设计、逻辑结构设计、物理结构设计。体验了系统的设计开发过程。很多书上写的很清楚,貌似看着也很简单,思路非常清晰。但需要自己想办法去设计一个系统的时候才发现其中的难度。如果没有自信,没有目标,没有信心就不可能把事情做好,当其他人都在迷茫的时候,自己一定要坚信目标。有一个这样的感觉就是课程设计学到的东西比一个学期都多。

参 考 文 献

- [1] 韩超, 梁泉 . Android 系统原理开发要点详解, 2010.1
- [2] 李宁. Android/OPhone 开发完全讲义, 中国水利水电出版社, 2010.6
- [3] (美) 埃克尔 陈昊鹏. Java 编程思想 (第四版), 机械工业出版社, 2007.6
- [4] E2ECloud 工作室. 深入浅出 GoogleAndroid, 人民邮电出版社, 2009.8
- [5] 余志龙, 王世江. GoogleAndroidSDK 开发范例大全, 人民邮电出版社, 2010.6
- [6] 张涛, 黄强, 毛磊雅. 一个基于 JSON 的对象序列化算法[J]. 计算机工程与应用, 2007
- [7] 沈美, 于翔. 浅析在 Android 系统中 JSON 和 GSON 的用法[J]. 移动应用 2014