

大数据学院

软件大型实验周

课程设计说明书

题目： 简易教育

学生姓名： 吴海董 学号： 1807004739

专 业： 物联网工程

指导教师： 于一、柴锐、秦品乐、武瑞娟

2021 年 6 月

目录

1	设计的主要内容.....	1
1.1	设计目的.....	1
1.2	设计要求.....	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	服务端.....	1
2.2	硬件环境	1
3	课题设计	1
3.1	需求分析.....	2
3.1.1	项目背景	2
3.1.2	用户需求	2
3.1.3	可行性分析	2
3.1.4	项目概要	2
3.1.5	系统功能结构	3
3.2	概要设计.....	3
3.2.1	数据库概念设计	3
3.2.2	数据库逻辑结构设计	6
3.3	详细设计.....	10
3.3.1	功能描述	10
3.3.2	系统流程图	10
3.4	编码.....	11
3.4.1	登录注册模块.....	11
3.4.2	课程模块.....	16
3.4.3	班群模块.....	20
3.4.4	成绩分析模块	29

4	测试	31
4.1	测试用例设计	31
4.1.1	登陆测试模块	31
4.1.2	注册功能测试模块	32
4.1.3	加入社群测试模块	33
5	心得体会	35
	参 考 文 献	36

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。

1.2 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 平台的在线直播教育类 APP。

1 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 30、Gradle 版本为 4.6。

2.1.2 服务端

后台服务器基于 SpringBoot 框架、Mybatis 框架以及 MySQL 数据库设计与开发。主要使用 Java 语言进行逻辑编写、SQL 语言进行数据库操作。使用 JetBrains 公司的 IDEA 编译器进行编码。同时使用 Navicat For Mysql 可视化数据库软件。

2.2 硬件环境

Android 前端使用 Redmi K20 Pro Premium 进行测试，使用联想 ThinkPad E580 电脑进行开发。

3 课题设计

3.1 需求分析

3.1.1 项目背景

随着互联网大力发展，以及我国公民对教育的越来越重视，但随着受教育人数越来越多，教育资源有限，使得线下教育资源匮乏，因此线上教育类 App 便应运而生。

线上教育类 App，增加了学习上课的多样性，通过互联网的传播快、普及程度高等特征使得线上教育节省了大量的教育资源。使得更多人有了接受教育的机会。同时，线上教育相比于线下教育的繁杂性，用户只需要简单的在手机上操作，便可以进行上课、提交作业等，同时用户也可以随时随地的进行学习。

较比传统教育的单一化课程来说，在线课程更为多元化，可以有多种课程体系同时运营管理，课程设计更多样化，学员可根据自己的特色性学习需求合理选择在线课程。此外，在线课程能紧跟时代更新速度，及时优化课程体系。

3.1.2 用户需求

方便操作，界面良好，功能齐全，网络延迟低。

3.1.3 可行性分析

本项目是基于 Android、SpringBoot、Mybatis、MySQL 以及 MVC 设计模式进行开发的，整个 APP 分为客户端、服务端以及 Web 端。客户端使用 Android 实现，服务端使用 SpringBoot+Mybatis 框架进行开发。

在 Android 端的直播功能中，我采用了市面上成熟的腾讯直播 SDK，极大地提高了开发的效率。同时 Android 端界面采用了扁平化界面设计，极大的提高了用户体验感，并且采用设计美学原理来设计功能以及界面颜色。相比市面上类似 App 的功能繁杂、界面简陋等缺点，该款 App 极大的符合用户需求，同时简单上手的操作，也是该款 App 的特点。

在服务端开发中，SpringBoot+Mybatis 的组合极大的提高了开发的效率，同时主流的 SpringBoot 框架有着非常稳定的性能以及简单、易上手的功能。持久层 Mybatis 框架更是抛弃了繁杂的 JDBC 操作，使对数据库的操作更加简单、安全。

3.1.4 项目概要

简易教育是一款在线直播教育类 App，用户可以在该 App 上面进行观看直播

学习，同时该款 App 设有教师类与学生类两类用户，教师类用户在该 App 上面进行直播上课、发布作业、查看成绩分析、创建社群。学生类用户可以在该 App 上面进行观看直播学习、给课程打分、课程评论、加入班课、查看作业、提交作业、注册登录、修改信息、查看成绩分析等功能。本款线上教育 App 解决了因疫情在家无法上课的情况，教师与学生可以使用该款 App 进行线上教育，同时也增加了学习的灵活性与方便性。

3.1.5 系统功能结构

该系统主要包括注册登录、直播发起观看、修改信息、评论课程、课程评分、成绩发布、创建班课、提交作业、成绩分析、修改信息等功能。

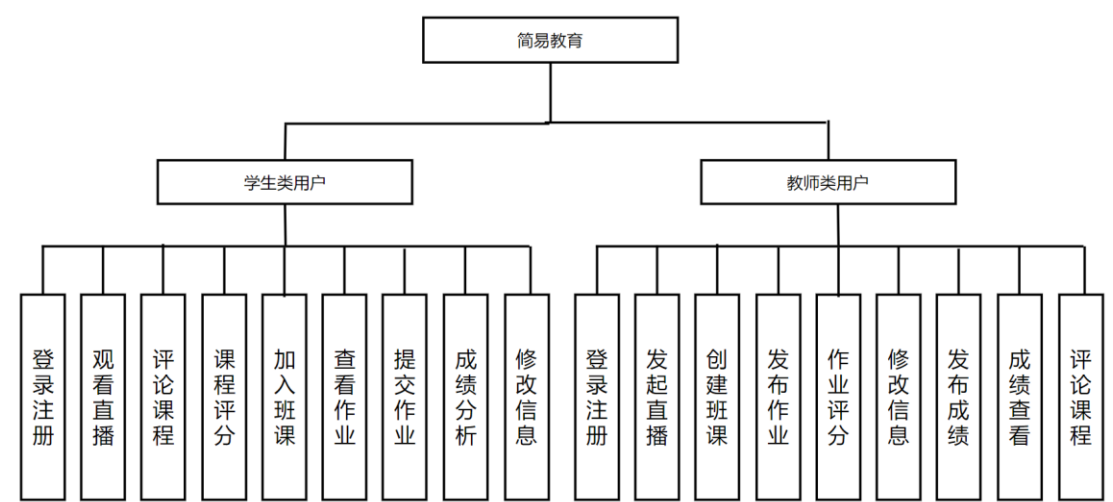


图 3.1 整体功能结构图

3.2 概要设计

3.2.1 数据库概念设计

简易教育 APP 数据库中共含有 12 个数据表，分别是学生表、教师表、课程表、社群表、作业表、作业文件表、学生进社群表、评论表、课程安排、学生选课表、学生评课表、学生提交作业表。对应的 E-R 图如下所示。

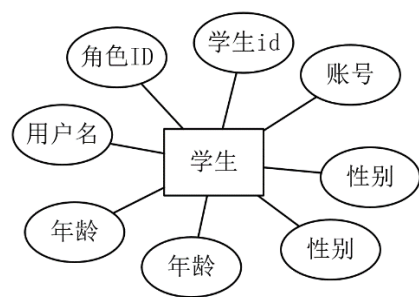


图 3.2 学生表 E-R 图

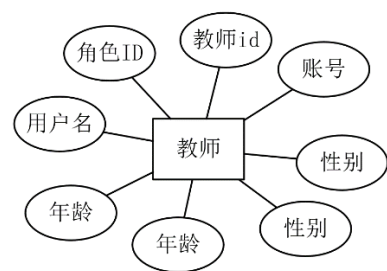


图 3.3 教师表 E-R 图

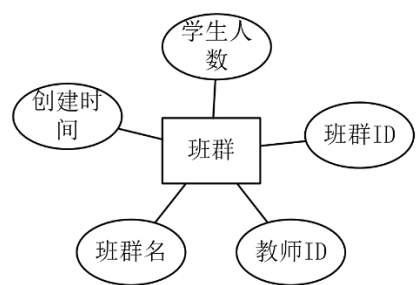


图 3.4 班群表 E-R 图

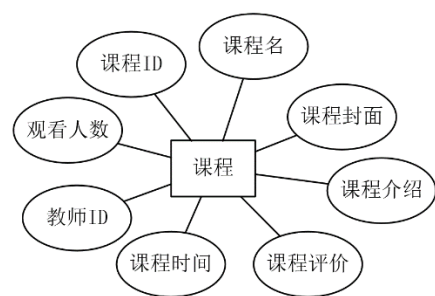


图 3.5 课程表 E-R 图

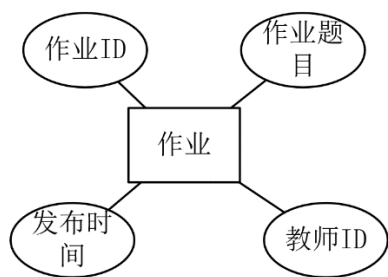


图 3.6 作业表 E-R 图

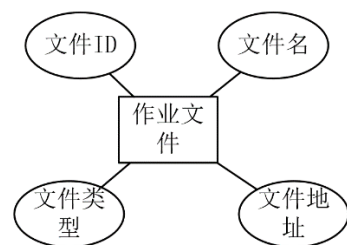


图 3.7 课程表 E-R 图

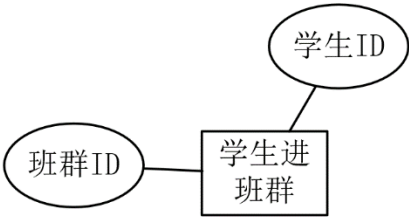


图 3.8 学生进班群 E-R 图

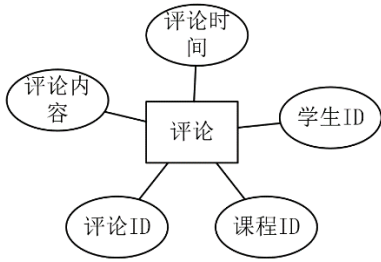


图 3.9 评论表 E-R 图

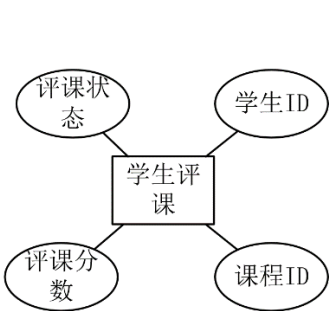


图 3.10 学生评课 E-R 图

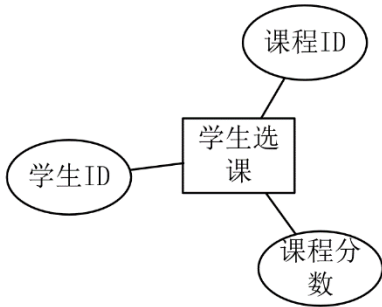


图 3.11 学生选课 E-R 图

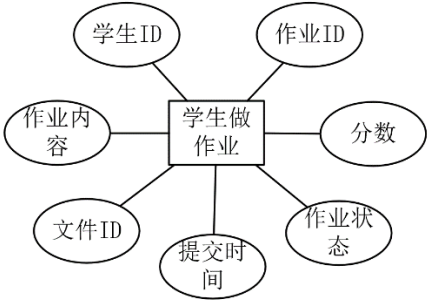


图 3.12 学生做作业 E-R 图

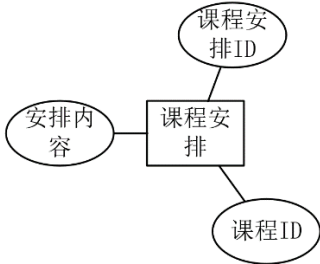


图 3.13 课程安排课 E-R 图

的主键转为字段。根据图 3.14 系统 E-R 图可分为 12 个数据库表。如下所示：

表 3.1 学生类用户表(student)

列名	数据类型	说明	是否为空
Id	int	学生 ID（主键）	否
Account	char	账号（主键）	否
Password	char	性别	
Sex	Int	类型	
telephone	char	班级号	
age	char	密码	
Name	char	用户名	
role	int	角色	

表 3.2 教师类用户表(teacher)

列名	数据类型	说明	约束
Id	int	教师 ID（主键）	否
Account	char	账号（主键）	否
Password	char	性别	
Sex	Int	类型	
telephone	char	班级号	
age	char	密码	
Name	char	用户名	
role	int	角色	

表 3.3 学生选课表(choose_course)

列名	数据类型	说明	是否为空
socre	float	成绩	
StudentID	int	学生 id（外键）	否
CourseID	int	课程 id（外键）	否

表 3.4 课程表(course)

列名	数据类型	说明	是否为空
ID	int	课程 id（主键）	否
name	varchar	课程名	否
introduce	varchar	课程介绍	否
evaluation	int	课程评分	否
time	varchar	内容	否
teacherId	Int	教师 id（外键）	否
view_number	Int	观看人数	否
course_url	varchar	课程封面地址	否

表 3.5 评论表(comment)

列名	数据类型	说明	是否为空
ID	int	评论 ID（主键）	否
student_id	int	学生 ID（外键）	否
course_id	int	课程 ID（外键）	否
content	varchar	评论内容	否
time	varchar	评论时间	否

表 3.6 作业(work)

列名	数据类型	说明	是否为空
ID	int	作业 id（主键）	否
Topic	varchar	作业题目	否
time	varchar	发布时间	否
teacherId	int	教师 id（外键）	否

表 3.7 学生进班群(enter_class)

列名	数据类型	说明	是否为空
Classid	Int	班群 id（主键）	否
Studentid	Int	学生 id（外键）	否

表 3.8 课程安排表(course_arrangement)

列名	数据类型	说明	是否为空
ID	int	安排 id (主键)	否
courseid	int	课程 id (外键)	否
content	varchar	安排内容	否

表 3.9 文件表(file)

列名	数据类型	说明	是否为空
Id	int	文件 ID (主键)	否
name	url	文件名	否
Type	Type	文件类型	否
url	url	文件地址	否

表 3.10 学生提交作业表(Student_work)

列名	数据类型	说明	是否为空
studentId	int	学生 id (外键)	否
workId	int	作业 id (外键)	否
Status	Int	提交状态	否
fileId	Int	文件 id (外键)	否
score	Float	作业分数	否
content	Varchar	作业内容	否
commit_time	varchar	提交时间	否

表 3.11 学生评分课程(course_score)

列名	数据类型	说明	是否为空
studentId	int	学生 ID (主键)	否
courseID	int	课程 ID (主键)	否
score	float	学生分数	否
status	Int	是否评分	否

3.12 班群表(choose_score)

列名	数据类型	说明	是否为空
ID	int	班群 ID（主键）	否
Teacherid	int	教师 ID（外键）	否
name	varchar	班群名	否
time	varchar	创建时间	否
Student_number	int	班群人数	否

3.3 详细设计

3.3.1 功能描述

简易教育具体包含如下功能：

- 学生登录注册
- 学生观看直播
- 学生评论课程、给课程打分
- 学生加入班课
- 学生查看作业列表
- 学生提交作业
- 学生查看成绩
- 学生修改信息

3.3.2 系统流程图

系统整体流程图如图 3.15 所示。

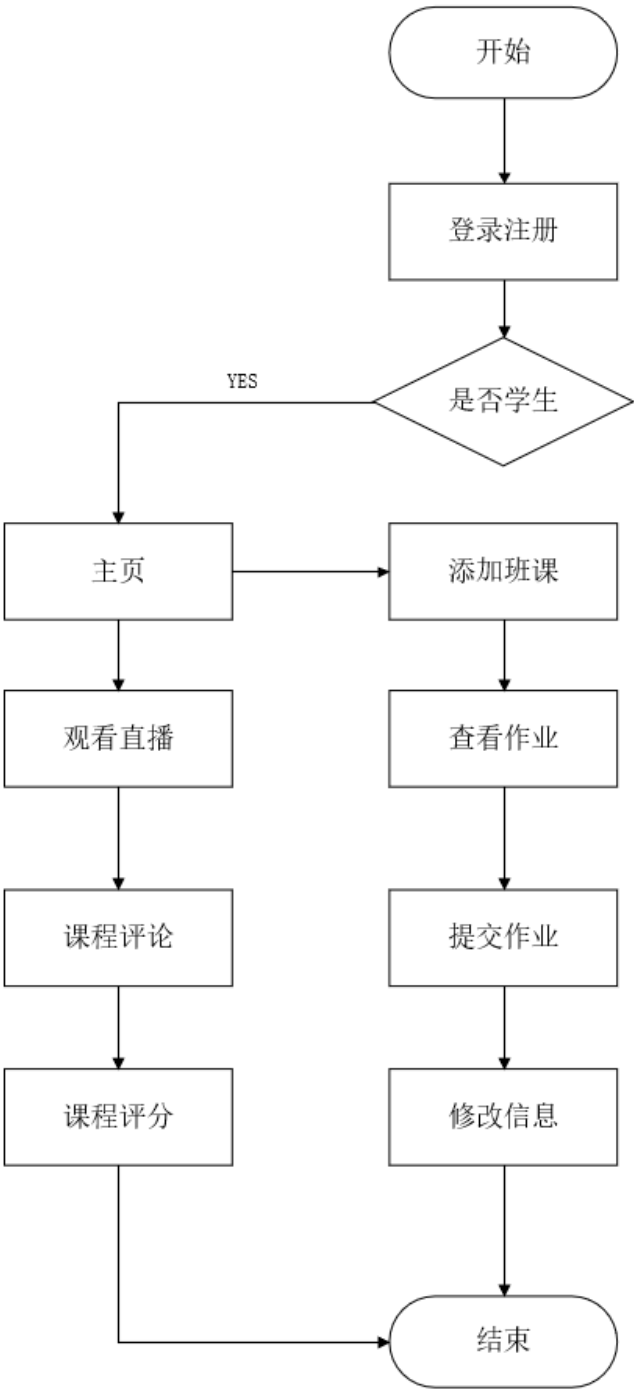


图 3.15 系统流程图

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 3.16 所示。

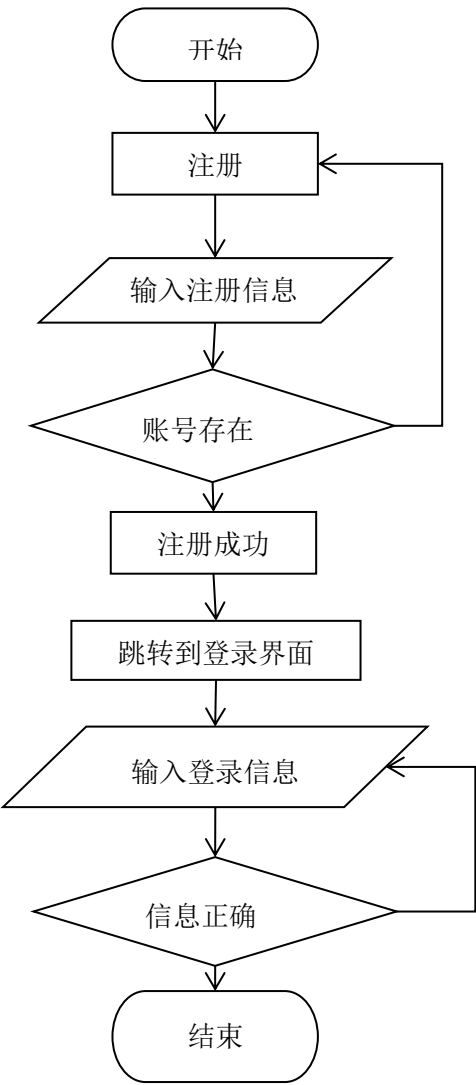


图 3.16 登录注册模块流程图

(3) 界面设计

登录和注册界面设计如图 3.17, 3.18 所示。



图 3.17 登录模块界面



图 3.18 注册模块界面

(4) 主要代码实现

```
private void postLogin(String json) {
    OkHttpUtil.usePost(url+"login", json).enqueue(new Callback() {
        @Override
        public void onFailure(Call call, IOException e) {
            httpResultInfo.setMsg("网络错误，请连接网络！");
            runOnUiThread()->{

                ToastUtil.show(httpResultInfo.getMsg(), LoginActivity.this);
            }
        }

        @Override
```



```

        public void onResponse(Call call, Response response)
throws IOException {
            String result = response.body().string();
            httpResultInfo = (HttpResultInfo)
JSONObject.parseObject(result, HttpResultInfo.class);

            if(httpResultInfo.getMsg().equals("登录成功")){
                String userString =
JsonUtil.objectToJson(httpResultInfo.getData());
                StudentInfo =
JSONObject.parseObject(userString, StudentInfo.class);
                Intent intent = new
Intent(LoginActivity.this, MainPageActivity.class);
                startActivity(intent);
                finish();
                runOnUiThread()->{

ToastUtil.show(httpResultInfo.getMsg(), LoginActivity.this);

                });
            }else{
                runOnUiThread()->{

ToastUtil.show(httpResultInfo.getMsg(), LoginActivity.this);

                });
            }
        });
    }

    studentInfo = new StudentInfo();

```

```
        studentInfo.setRole(1);

studentInfo.setPassword(password.getText().toString());

studentInfo.setTelephone(telephone.getText().toString());

studentInfo.setAccount(editTextUser.getText().toString());

studentInfo.setName(editTextUserName.getText().toString());

studentInfo.setAge(Integer.valueOf(age.getText().toString()));
        if(boy){
            studentInfo.setSex("男");
        }else{
            studentInfo.setSex("女");
        }
        new Thread(new Runnable() {
            @Override
            public void run() {

postRegister(JsonUtil.objectToJson(studentInfo));
                if(httpResultInfo.getMsg().equals("注册成功")){

                    finish();
                    runOnUiThread()->{

ToastUtil.show(httpResultInfo.getMsg(), RegisterActivity.this);
                        });
                    }else{
```

```

runOnUiThread(()->{

ToastUtil.show(httpResultInfo.getMsg(),RegisterActivity.this);

});

}

}).start();
    
```

3.4.2 课程模块

(1) 功能描述

学生用户进入课程进行观看直播学习,同时可以对直播课程进行评论、评分。

(2) 流程图

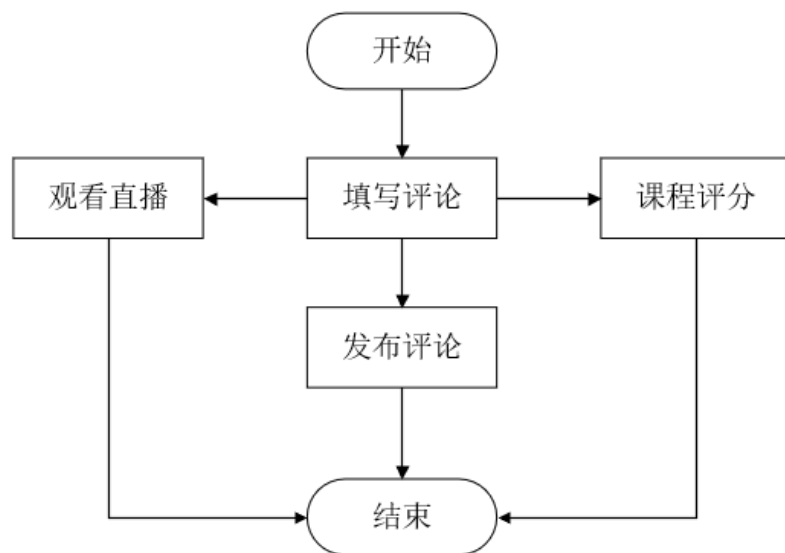


图 3.19 课程模块流程图

(3) 界面设计



图 3.19 课程列表界面



图 3.20 课程主界面

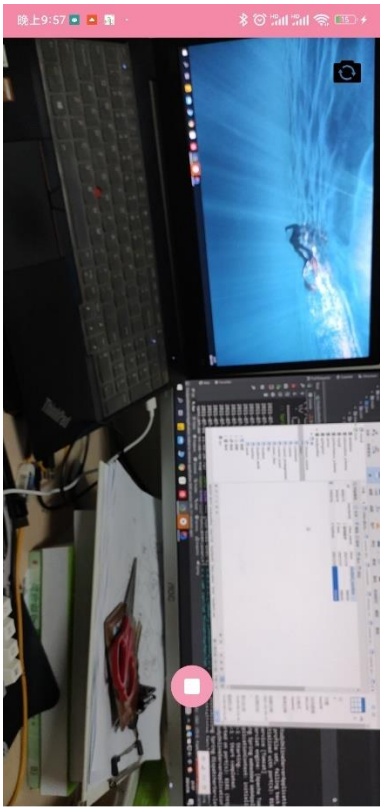


图 3.21 直播界面



图 3.22 评论界面



图 3.23 评分成功界面



图 3.24 课程安排界面

(4) 主要代码实现

```
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);
    }

    @Nullable

    @Override

    public View onCreateView(LayoutInflater inflater, @Nullable
    ViewGroup container, Bundle savedInstanceState) {

        View view =
inflater.inflate(R.layout.fragment_courselive, container, false);

        listView = view.findViewById(android.R.id.list);
        courseInfoArrayList = new ArrayList<>();
        OkHttpUtil.usePostById(url+"/courses",
LoginActivity.studentInfo.getId()).enqueue(new Callback() {

            @Override

            public void onFailure(Call call, IOException e) {

                System.out.println(e);

            }

            @Override

            public void onResponse(Call call, Response response)
throws IOException {

                String result = response.body().string();

                courseInfoArrayList = (ArrayList<CourseInfo>)
JSONObject.parseArray(result, CourseInfo.class);

                getActivity().runOnUiThread(new Runnable() {

                    @Override

                    public void run() {

                        courseListAdapter = new

```

```
CourseListAdapter(getActivity(), courseInfoArrayList);  
                listView.setAdapter(courseListAdapter);  
            }  
        });  
    }  
});  
    return view;  
}  
  
@Override  
    public void onItemClick(@NonNull ListView l, @NonNull View v,  
int position, long id) {  
        super.onItemClick(l, v, position, id);  
        CourseInfo courseInfo = courseInfoArrayList.get(position);  
        Intent intent = new Intent(getActivity(),  
CourseInfoActivity.class);  
        intent.putExtra("name", courseInfo.getTeacherName());  
        intent.putExtra("id", courseInfo.getId());  
  
        startActivity(intent);  
    }  
}
```

3.4.3 班群模块

(1) 功能描述

用户加入班群，并进入班群页面之后查看作业列表，没有提交的作业点击进入作业页面，提交作业。

(2) 流程图

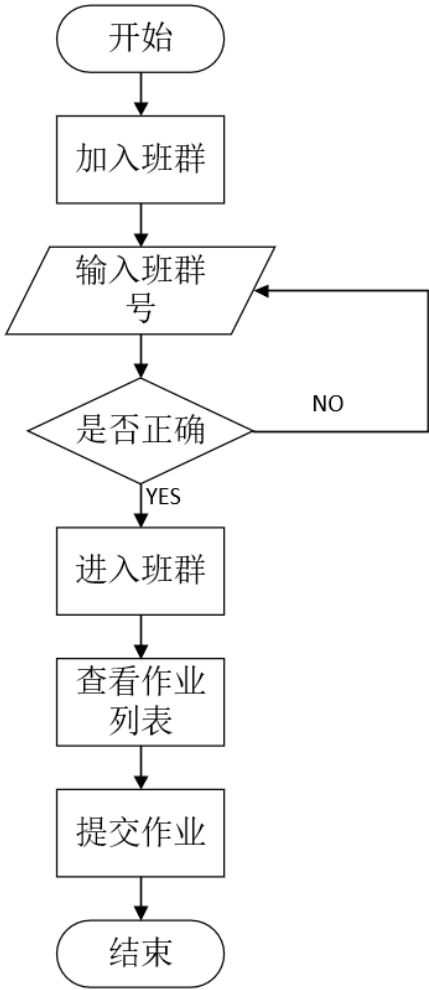


图 3.25 课程模块流程图

(3) 界面设计



图 3.26 加入班课界面



图 3.27 加入班课成功界面



图 3.28 作业列表界面



图 3.29 提交作业界面

(4) 主要代码实现

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_write_work);  
  
    if (Build.VERSION.SDK_INT > Build.VERSION_CODES.LOLLIPOP) {  
        if (ActivityCompat.checkSelfPermission(this,  
Manifest.permission.WRITE_EXTERNAL_STORAGE) !=  
PackageManager.PERMISSION_GRANTED) {  
            ActivityCompat.requestPermissions(this,  
PERMISSIONS_STORAGE, REQUEST_PERMISSION_CODE);  
        }  
    }  
  
    initView();  
    textView.setOnClickListener(this);  
    commit.setOnClickListener(this);  
  
}
```

@Override

```
public void onClick(View v) {  
    switch (v.getId()) {  
        case R.id.find_file:  
            Intent intent = new Intent(Intent.ACTION_GET_CONTENT);  
            intent.setType("*/*");//无类型限制  
            intent.addCategory(Intent.CATEGORY_OPENABLE);  
            startActivityForResult(intent, 33);  
            break;
```

```
        case R.id.commit:
            if(editText.getText().toString().equals("")){
                ToastUtil.show("  内  容  不  能  为  空  ！
",WriteWorkActivity.this);
            }else{

workInfo.setContent(editText.getText().toString());

                SimpleDateFormat tempDate = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

                String datetime = tempDate.format(new
java.util.Date());

                workInfo.setCommitTime(datetime);
                workInfo.setStatus(1);
                commitWork();
            }
            break;

    }
}
```

```
private void commitWork() {
    String json = JsonUtil.objectToJson(workInfo);
    OkHttpUtil.usePost(url, json).enqueue(new Callback() {
        @Override
        public void onFailure(Call call, IOException e) {
            System.out.println(e);
        }
    })
}
```

```

        @Override

        public void onResponse(Call call, Response response)
throws IOException {

            String result= response.body().string();

            resultInfo                                =
JSONObject.parseObject(result,HttpResultInfo.class);

            if(resultInfo.isSuccess()){

                runOnUiThread(new Runnable() {

                    @Override

                    public void run() {

                        ToastUtil.show("    提    交    成    功

",WriteWorkActivity.this);

                    }

                });

                finish();

            }

        }

    });

}

```

```

    @Override

    protected void onActivityResult(int requestCode, int resultCode,
Intent data) {

        super.onActivityResult(requestCode, resultCode, data);

        if (resultCode == Activity.RESULT_OK) {

            if (requestCode == 33) {

                Uri uri = data.getData();

                getFileInfo(uri);

            }

        }

    }

```

```

        }
    }
}

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);

    if (requestCode == REQUEST_PERMISSION_CODE) {
        for (int i = 0; i < permissions.length; i++) {
            Log.i("MainActivity", "申请的权限为: " + permissions[i]
+ ", 申请结果: " + grantResults[i]);
        }
    }
}

private void getFileInfo(Uri uri){

    Cursor returnCursor = getContentResolver().query(uri, null,
null, null, null);

    int nameIndex =
returnCursor.getColumnIndex(OpenableColumns.DISPLAY_NAME);

    int sizeIndex =
returnCursor.getColumnIndex(OpenableColumns.SIZE);

    returnCursor.moveToFirst();

    String fileName = returnCursor.getString(nameIndex);

```

```
workInfo.setFileName(fileName);
workInfo.setUrl(uri.getPath());

String[] token = fileName.split("\\.");
String s = token[1];

String type = s.toLowerCase();
workInfo.setFileType(type);

if (type.equals("doc") || type.equals("docx")) {

imageView.setImageDrawable(getResources().getDrawable(R.drawable.word
));
    }else if(type.equals("txt")){

imageView.setImageDrawable(getResources().getDrawable(R.drawable.txt)
);
    }else if(type.equals("ppt") || type.equals("pptx")){

imageView.setImageDrawable(getResources().getDrawable(R.drawable.ppt)
);
    }else if(type.equals("pdf")){

imageView.setImageDrawable(getResources().getDrawable(R.drawable.pdf)
);
    }else if(type.equals("xls") || type.equals("xlsx")){

imageView.setImageDrawable(getResources().getDrawable(R.drawable.exce
```

```
1));  
  
    }else{  
        imageView.setImageURI(uri);  
    }  
    workName.setText(fileName);  
  
    fileSize.setText(longToByteSize(returnCursor.getLong(sizeIndex)));  
    linearLayout.setVisibility(View.VISIBLE);  
}  
  
private String longToByteSize(long size){  
    if(size == 0){  
        return null;  
    }  
    if (size < 1024) {  
        return String.valueOf(size) + "b";  
    } else {  
        size = size / 1024;  
    }  
    if (size < 1024) {  
        size = size * 100;  
        return String.valueOf((size / 100)) + "."  
            + String.valueOf((size % 100)) + "Kb";  
    } else {  
        size = size * 100 / 1024;  
        return String.valueOf((size / 100)) + "."  
            + String.valueOf((size % 100)) + "Mb";  
    }  
}
```

```
}
```

3.4.4 成绩分析模块

(1) 功能描述

学生可以在成绩分析页面查看自己的成绩，同时与班级最高分最低分，班级平均分做比较

(2) 流程图

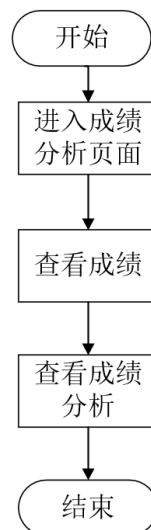


图 3.30 成绩分析模块流程图

(3) 界面设计

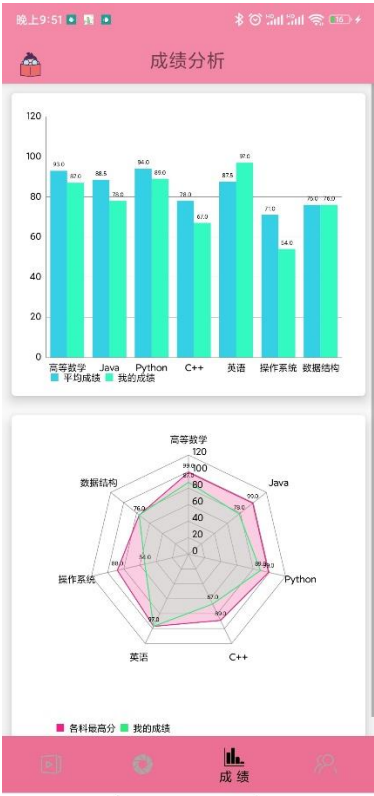


图 3.31 成绩分析界面

(4) 主要代码实现

```
@Nullable
@Override
public View onCreateView(LayoutInflater inflater, @Nullable
    ViewGroup container, Bundle savedInstanceState) {
    View view =
inflater.inflate(R.layout.fragment_score_analysis, container, false);
    initView(view);
    courseScoreAnalysisInfo = new CourseScoreAnalysisInfo();
    getScoreData();
    barChartUtil = new BarChartUtil(barChart);
    return view;
}
```

```
private void getScoreData() {  
    OkHttpUtil.usePostById(CourseLiveFragment.url+"/score",  
LoginActivity.studentInfo.getId()).enqueue(new Callback() {  
        @Override  
        public void onFailure(Call call, IOException e) {  
            System.out.println(e);  
        }  
  
        @Override  
        public void onResponse(Call call, Response response)  
throws IOException {  
            String result = response.body().string();  
            courseScoreAnalysisInfo =  
JSONObject.parseObject(result, CourseScoreAnalysisInfo.class);  
            getActivity().runOnUiThread(new Runnable() {  
                @Override  
                public void run() {  
  
barChartUtil.showBarChart(courseScoreAnalysisInfo.getMyScore(), course  
ScoreAnalysisInfo.getAverageScore());  
  
showRadarChart(courseScoreAnalysisInfo.getMyScore(), courseScoreAnalys  
isInfo.getMaxScore());  
                }  
            });  
        }  
    });  
}
```

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
123	123	成功	成功
123	123456	密码错误	密码错误
123456789	123456789	用户名不存在	用户名不存在

登陆测试结果如图 4.1、4.2、4.3 所示。

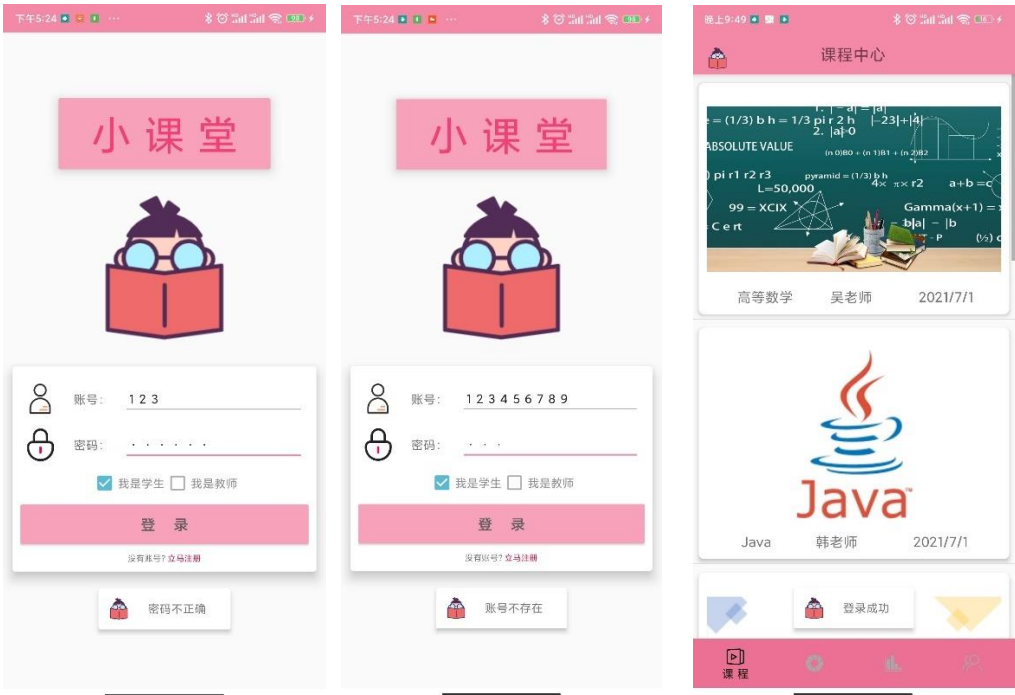


图 4.1 登录测试用例 1 图 4.2 登录测试用例 2 图 4.3 登录测试用例 3

4.1.2 注册功能测试模块

注册用户时可能会出现以下情况：账号已被注册，输入密码不对，注册成功等情况，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 注册测试用例表

用户名	密码	再次输入密码	预期结果	实际结果
123	123	123	错误	账户已被注册

123456	123456	123	错误	密码不一致
123456	123456	123456	成功	成功

注册测试结果如图 4. 4、4. 5、4. 6 所示。



图 4. 4 注册测试用例 1



图 4. 5 注册测试用例 2



图 4. 6 注册测试用例 3

4. 1. 3 加入社群测试模块

学生类用户通过输入社群号加入社群可能出现已经加入社群、社群号不存在、加入成功的情况。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. 3 所示。

表 4. 3 加入社群测试用例表

邀请码	预期结果	实际结果
543216	成功	加入社群成功
898578	错误	已存在班课
1	错误	社群不存在

加入社群测试结果如图 4. 7、4. 8、4. 9、4. 10、4. 11、4. 12 所示。



图 4.7 社群测试用例 1



图 4.8 社群测试用例 2



图 4.9 社群测试用例 3



图 4.10 社群测试用例 4



图 4.11 社群测试用例 5



图 4.12 社群测试用例 6

5 心得体会

通过本次 Android 课程设计，我深深的了解到了自身能力的不足以及知识的浅薄，在本次课程设计过程中，因为自己基础不牢而出现了很多问题，真正的体现了：基础不牢，地动山摇这句话。不过，通过此次课程设计，我也收获了很多东西，粗略的了解了软件开发的整个流程，同时对书本上的知识有了更深刻的理解，并将自己所学到的东西真正运用到实践上。

本次 Android 课设，让我懂得了耐心以及细心在编程过程中的重要性，当我们遇到问题的时候，不应该急躁，而是慢慢的去解决问题，办法总比问题多，只要我们细心、仔细，bug 一定会被解决的。我也发现，自己在开发过程中过度追求界面 UI，从而使真正开发项目时间大大减少，这是我的一个不好的小毛病。做事情不能顾此失彼，要有效的结合自己的能力去做事情，不能钻牛角尖。

在此次课程设计过程中最大的收获是了解了服务端与前端的通信问题，将大二所学的计算机网络真正应用到了实践之中。在加深 Android 开发的过程中，也顺带复习了上学期所学的 Springboot 后端开发框架。

最后，由于时间关系，本项目并没有实现太多功能，仅仅实现了一些基础功能，留下了小小的遗憾，但是虽然课程设计结束了，但是我的学习之路并没有结束，是课程设计的终点，也是我迈进下一步学习的起点。

参 考 文 献

- [1] 基于 Android 移动学习平台的设计与实现[J]. 张济先. 现代工业经济和信
息化. 2019(03)
- [2] 基于 Android 的移动学习平台的研究与设计[J]. 韩建林. 电子技术与软件
工程. 2018(04)
- [3] 基于 Android 的移动学习平台研究与设计[J]. 张新猛. 福建电脑. 2016(06)
- [4] 宋艳 . 基于 Android 系统的交互式移动学习平台设计 [J]. 通讯世
界 , 2019(11):82.
- [5] 吴明瑞 , 王莉军 . 基于 Android 操作系统的移动终端 APP 开发研究
[J]. 数字通信世界 , 2019(2):113.
- [6] 胡剑锋 . 基于 Android Studio 平台的图书阅读器的设计与 实现 [J].
电脑知识与技术 , 2018, 14(36):83-84.
- [7] 李晓康 , 王成田 , 张德青 . Android 平台下“易书”APP 的设计与实现
[J]. 电子商务 , 2018(5):45-46.