题目

基于图像识别技术的数学教辅APP的设计与实现

**摘要**

    教育问题是学生家长一直比较重视的问题，伴随着互联网的发展，教育问题大多都已经线上化，在加上今年疫情的影响，导致线上教育更是非常火热，这也是移动互联网和产业互联网的融合，做好线上教育，充分发挥家长和老师在线上的最大作用，是当前非常关注的一个问题。要将线上线下融合，就需要充分理解线下教育，将整个链路数字化实现，包括学生在学校的表现，以及学生的学习习惯等。

    该APP拥有组建班级功能，可以让家长和老师进行充分的互动，实现线上线下作业，相比于传统的微信沟通，更加标准化。基于图像识别技术，可以识别学生以及家长的题目，进行四节运算的判断。通过图片校验和线上练习，可以更加明确学生在学习方面的错误和短板，加强错题的推送，让学生在错误中成长。对于同学的日常检测和练习记录，包括对应的错题记录，都会存历史，保证学习生可以温故而知新。

    针对于目前家长关注的教育问题，该产品进行竞品分析和需求调研，让家长和老师可以通过线上互动，来管理孩子的学习。家长也可以轻松的参与到孩子的学习之中。从小抓好学生教育，对孩子的成长都有莫大的作用，意义非凡。经过严格的测试，项目已经可以稳定提供服务，服务已成功部署到云端，可以直接在网络正常的环境下进行访问。

关键词：APP；图像识别；产业互联网；移动互联网

**ABSTRACT**

    Education is students' parents have been more attention problems, with the development of the Internet, education problem mostly have online, plus the influence of the outbreak this year, lead to the online education is very hot, that is the integration of the Internet, mobile Internet and business for online education, give full play to their parents and teachers are the biggest online, is a problem concerned by the current. To integrate online and offline education, it is necessary to fully understand offline education and digitize the whole link, including students' performance in school and their learning habits.

    This APP has the function of class building, which enables parents and teachers to interact with each other fully and realize online and offline homework. Compared with the traditional WeChat communication, it is more standardized. Based on the image recognition technology, students and their parents can be identified and judged by four sections of calculation. Through picture check and online practice, students can be more clearly in learning errors and shortcomings, to strengthen the push of wrong questions, so that students grow up in the mistakes. Students' daily test and practice records, including the corresponding error records, will be stored in the history, to ensure that students can review the old and learn the new.

    Aiming at the current educational issues concerned by parents, this product conducts competitive product analysis and demand research, enabling parents and teachers to manage children's learning through online interaction. Parents can also easily participate in their children's learning. Pay attention to student education since childhood, have great effect to the growth of the child, meaning is special. After rigorous testing, the project has been able to provide the service stably, the service has been successfully deployed to the cloud, can be accessed directly in the normal network environment.

Key words: APP; Image recognition; Industrial Internet; Mobile Internet

**第一章 绪论**

**1.1 课题背景**

    随着国家的发展，我国教育水平已经做的非常到位，比如国家的九年义务教育，可以让受教育的覆盖面达到百分之百，但是事实上做到九年义务教育已经不够了。在现在大形势下，互联网的发展，很多产品已经从线下转移到线下，当然对于教育行业， 肯定不能舍弃线下，但是我们可以将线上线下融合，让家长也参与进去，促进学生的学习。并且从今年开年的疫情来看，各个学校采用线上教学的方式，通过家长和老师配合促进学生学习。根据我的调研情况，我们会发现这个方案是非常有价值的。

    要进行线上线下融合，就要充分利用当前的科学技术。随着科技的进一步发展，互联网技术体系已经发展的足够完全，在前几年，大多表现的是一种移动互联网营销的模式，在近几年，各个互联网公司尝试进行产业互联网的探索，将线上线下融合，实现链路数字化。再加上近几年机器学习、神经网络方面技术的几步，让图像识别技术更加准确。因此，尝试将最新的技术应用到产品中，发挥技术的最大价值，毕竟技术就是服务于业务，业务就是服务于人的生活。

    对于当前项目的开发， 业界已经又了非常成熟的技术方案，比如移动应用的开发，足够了解Android技术以及后台技术。对于后台开发，采用业界流行的SpringBoot框架，MVC三层架构，项目逻辑架构清晰。安卓采用原生的组件和布局，通过HTTP协议进行网络通信，服务端启动Tomact服务器。我们通过应用现有的技术，可以构建出来一个高可用的服务，让系统可以稳定的运行。对于开发完成，可以将服务发不到阿里云轻量级服务器，进行服务的部署。参考DevOps技术，实现项目的自动化部署。

**1.2 国内外研究现状**

    目前国内外针对图像识别还都是在研究探索阶段，虽然已经有了很大的成果，但是对于手写识别技术，还不是足够的成熟，应用也不是非常的广泛。技术需要落地，就需要在各个行业进行不断的探索，这正是我们需要做的事情。

    对于教辅APP，国内也有很多探索，但是大部分产品定位不明确。常见的问题比如：因为太多的功能导致家长不能聚焦到学生的学习重点、产品定位不明确等。很多产品已经偏离了本质的教育问题，并且产生一些不必要的增值服务。对于功能方面，国内大部分产品落地不是很完善，功能相对碎片化。

**1.3 课题研究目标**

    本课题主要是为了打破传统的线下教育的方式，通过线上和线下结合，让家长不仅参与到学生的生活中，更要参与到他们的学习中，通过线上APP的形式，家长可以更容易的去管理学生的学习，为学生进行学习指导。也方便老师和家长之间的沟通，快速解决学生面临的困难，让学生的学习不再孤单，实现三方共赢的目的。

**1.4 本文内容安排**

本文内容主要安排如下：

第一章绪论，主要介绍目前的项目背景，说明课题的价值和意义。

第二章系统需求，主要介绍该产品包含哪些功能。

第三章技术调研，主要介绍相关技术原理，对技术进行调研分析。

第四章系统设计，介绍项目如何进行设计，包括流程设计，需求实现，存储设计，物理架构和逻辑架构等。

第五章系统开发，主要说明开发存疑的问题，开发对应的环境已经开发成果。

第六章系统测试，主要介绍如何进行测试，测试流程。

第七章总结和展望，总结全文，并说明未来需要完善的点。

**第二章 系统需求**

**2.1 业务需求**

该项目主要面向的角色为老师和家长。具体业务流程如图2.1：

图2.1  业务操作流程图

    对于每一个用户，首先需要进行注册。注册成功之后即可登陆。登陆成功可以使用拍照检查、获取拍照历史、练习等功能。对于老师，可以创建班级、发布作业。家长可以加入老师创建的班级，并且可以获取到老师发布的作业信息。

**2.2 系统功能需求**

    系统功能需求如图2.2，其中主要包括不同身份（家长、老师）的功能描述。老师和家长共同的功能主要包括：登陆、注册账户、拍照检查作业、练习、查看班级作业、查看检查历史、查看错误历史、更新用户信息。老师特有的功能主要包括：创建班级、发布作业、管理班级用户、解散班级。家长特有的功能主要包括：加入班级、退出班级。

图2.2  业务用例图

**2.2.1 注册**

* 用例名称:用户注册。
* 概述:用户通过邮箱、密码、验证码进行账户注册。
* 主要角色:所有用户。
* 前置条件:用户在有网络的环境下打开客户端，点击注册按钮。
* 后置条件:注册成功，注册数据持久化到数据库。
* 主成功场景:
  + 用户注册成功。

**2.2.2 登陆**

* 用例名称:用户登陆。
* 概述:用户通过邮箱、密码进行账户登陆
* 主要角色:所有用户。
* 前置条件:用户在有网络的环境下打开客户端，点击登陆按钮。
* 后置条件:登陆成功。
* 主成功场景:
  + 用户登陆成功。

**2.2.3 拍照检查作业**

* 用例名称:用户拍照检查作业。
* 概述:用户打开相机拍照或者上传相册图片，检查图片作业。
* 主要角色:所有用户。
* 前置条件:用户在有网络的环境下打开客户端，并且登陆。
* 后置条件:检查成功，检查数据被存储到检查历史，错题记录错误历史。
* 主成功场景:
  + 检查成功。
  + 返回检查结果。

**2.2.4 练习**

* 用例名称:用户练习。
* 概述:用户切换到练习模块，输入练习数量，获取练习题目，提交练习结果。
* 主要角色:所有用户。
* 前置条件:用户在有网络的环境下打开客户端，并且登陆。
* 后置条件:练习成功，练习结果记录到练习历史和错误历史。
* 主成功场景:
  + 练习成功。
  + 返回练习结果。

**2.2.5 查看班级作业**

* 用例名称:查看班级作业。
* 概述:用户点击班级页面，加载所有班级作业。
* 主要角色:所有用户。
* 前置条件:用户在有网络的环境下打开客户端，并且登陆。
* 后置条件:获取作业成功，客户端展示所有结果。
* 主成功场景:
  + 返回作业列表。

**2.2.6 查看检查历史**

* 用例名称:查看检查历史。
* 概述:用户点击检查历史，查看所有检查历史。
* 主要角色:所有用户。
* 前置条件:用户在有网络的环境下打开客户端，并且登陆。
* 后置条件:获取历史成功，客户端展示练习历史和检查历史。
* 主成功场景:
  + 返回所有检查历史。

**2.2.7 查看错题历史**

* 用例名称:查看错题历史。
* 概述:用户点击错题历史，查看所有错题历史。
* 主要角色:所有用户。
* 前置条件:用户在有网络的环境下打开客户端，并且登陆。
* 后置条件:获取历史成功，客户端展示练习错题历史和检查错题历史。
* 主成功场景:
  + 返回所有错题历史。

**2.2.8 创建班级**

* 用例名称:创建班级。
* 概述:老师创建班级。
* 主要角色:老师。
* 前置条件:用户在有网络的环境下打开客户端，并且登陆。
* 后置条件:创建班级成功，持久化班级数据。
* 主成功场景:
  + 返回用户所有班级。

**2.2.9 移除班级成员**

* 用例名称:移除班级成员。
* 概述:老师选择想要移除的班级成员，进行移除。
* 主要角色:老师。
* 前置条件:用户在有网络的环境下打开客户端，并且登陆。
* 后置条件:移除成员成功，删除成员数据。
* 主成功场景:
  + 移除成功，返回剩余成员列表。

**2.2.10 发布作业**

* 用例名称:发布作业。
* 概述:老师选择班级，发布班级作业。
* 主要角色:老师。
* 前置条件:用户在有网络的环境下打开客户端，并且登陆。
* 后置条件:发布成功，作业数据持久化。
* 主成功场景:
  + 发布成功。
  + 返回作业列表。

**2.2.11 个人信息管理**

* 用例名称:个人信息管理。
* 概述:用户获取、更新个人信息。
* 主要角色:所有用户。
* 前置条件:用户在有网络的环境下打开客户端，并且登陆。
* 后置条件:修改个人信息，持久化信息。
* 主成功场景:
  + 修改成功，返回最新个人信息。

**2.2.12 加入班级**

* 用例名称:加入班级。
* 概述:用户输入班级编号，加入班级。
* 主要角色:家长。
* 前置条件:用户在有网络的环境下打开客户端，并且登陆。
* 后置条件:加入成功，持久化用户班级信息。
* 主成功场景:
  + 加入成功，返回加入的班级列表。

**2.3 系统非功能需求**

**2.3.1 界面需求**

    客户端界面使用安卓原生控件开发。

**2.3.2 性能需求**

    1.客户端操作顺畅，不能有卡顿现象。

    2.上传图片做优化，图片在服务端做缓存。

    3.对于频繁拉取的用户信息做缓存。

**第三章 技术调研**

**3.1 手写识别技术调研**

**3.2 四则运算实现**

经过调研，四则运算主要实现方式有两种，第一就是基于堆栈的四则运算算法，第二就是通过编译原理构造抽象语法树实现四则运算。

**3.2.1 堆栈算法**

**3.2.1.1 中缀表达式转后缀表达式**

    从左到右遍历中缀表达式的每一数字和符号，若是数字就输出，即成为后缀表达式的一部分；若是符号，则判断其与栈 顶符号的优先级，是右括号或优先级低于栈顶符号（乘除优先加减）则栈顶元素依次出栈并输出，并将当前符号进栈，一直到最终输出后缀表达式为止。

**3.2.1.2 后缀表达式计算结果**

    从左到右遍历表达式的每个数字和符号，如果当前字符是数字，那么就进栈，如果是符号，就将处于栈顶两个数字出栈，执行运算，运算结果进栈。重复上述操作，一直到最终获得结果。如果遇到非法字符或者栈不为空，那么直接做异常处理。

**3.2.2 编译原理四则运算编译器**

**3.2.2.1 词法分析**

1.总结数据中可能出现的Token，归纳如下：

* 字母 （ALPHABET）
* 左括号（LEFT\_PARENTHESES）
* 右括号（RIGHT\_PARENTHESES）
* 运算符：加（PLUS），减（SUBTRACT），乘（MULTIPLE），除（DIVISION）

2.根据Token进行词法的解析。

**3.2.2.2 语法分析**

1.构建语法规则

2.消除歧义

3.左结合

4.递归向下算法

5.构建AST

6.根据AST计算结果

**第四章 系统设计**

**4.1 设计决策**

    技术方案：如图4.1所示，整个系统主要包括两个模块：客户端、后端。对于客户端采用安卓原生开发，基于Android SDK实现客户端开发。对于后端开发，采用业界流行的SpringBoot，类似微服务开发模式。后端项目采用MVC三层架构，依赖SpringMVC、Tomcat服务器实现Web开发。客户端和后端采用HTTP协议通信，主要访问形式分为表单和JSON。数据库依赖Redis和Mysql。Mysql主要做数据持久化，Redis做数据缓存。整个技术架构清晰，前后端分离，降低耦合度。

    工具：Android Studio、Android SDK、JDK、IDEA、Maven、Gradle

    环境平台：Mac 10.14.6（开发）、安卓真机（调试）、安卓虚拟机（调试）、阿里云CentOS服务器（服务部署）

图4.1  开发视图

**4.2 体系结构设计**

**4.2.1 逻辑架构**

    整体架构如图4.2，其中客户端主要就是Android原生实现，安卓通过HTTP和后端服务器进行数据交换。安卓请求被传递到后端，后端通过SpringMVC转发到对应的控制层处理器，控制层调用业务逻辑层执行对应的业务处理，处理过程中会将必要的数据持久化到数据库或者存储到Redis中，然后原路返回。

图4.2  逻辑架构图

**4.2.2 开发架构**

开发工程目录如图4.3、4.4所示。

图4.3  安卓目录结构

具体目录解释如下：

Fragment主要就是一些页面碎片，嵌套在Activity之上。java目录主要就是一些业务实体。Activity为对应页面的活动逻辑。anim存放dialog的进出动画，drawable存放进出动画，layout为安卓的页面布局的xml文件。values为安卓的样式文件。

图4.4  后端目录结构

具体目录解释如下：

aspect为SpringAOP代理，用于全局拦截，日志记录。bean为java的实体VO，calexpr为四则运算处理器。common为项目公共实现。config为Spring的配置类。Controller为处理HTTP接口的处理器。Dao为持久化层。entity为数据库实体。ocr为图像识别的接口。service为业务逻辑实现类。

**4.2.3 物理架构**

    如图4.5，用户通过客户端路由到服务器防火墙，然后转到对应的Web服务器，Web服务器通过局域网和数据库服务器建立通信。

图4.5  物理架构图

**4.3 数据存储设计**

**4.3.1 数据表结构**

    项目使用Mysql进行存储，主要表结构相对简单，如图4.6所示

图4.6  数据库表结构图

**4.3.2 数据表介绍**

    数据库采用Mysql的Innodb存储引擎，每个表都有一个字增的主键id。user表主要存储用户的详细信息。class表为班级信息表。school表存储所有学校信息。check\_history和error\_history分别存储检查历史和错题历史。home\_work存储老师布置的作业信息。class\_user为用户和班级的关系表。

**4.5 业务模块设计**

**4.5.1 用户模块设计**

    用户模块功能比如注册、登陆、用户信息管理功能。

**4.5.1.1 软件单元构成**

    用户模块对应的类图如图4.7，其中UserServiceImpl实现了UserService接口。客户端的用户相关操作都会UserController提供。UserServiceImpl依赖MailService、RedisDao、UserDao完成对应的业务功能。MailService主要用于注册时发送验证码。RedisDao用于存储用户的验证码。UserDao主要将用户信息持久化到数据库。

图4.7  用户模块类图

**4.5.1.2 用户注册流程**

    大概流程如图4.8，用户点击安卓注册按钮，会跳到注册页面，用户输入邮箱和密码信息，然后点击获取验证码按钮，如果未输入邮箱或者输入错误邮箱，会直接提示输入正确的邮箱地址。后台UserService会进行邮箱校验，如果邮箱已经存在，直接返回该邮箱已经被注册。如果不存在，则通过MailService进行验证码发送和存储。

    用户端收到邮箱验证码后，填入验证码，执行提交操作。点击提交按钮，UserService会进行邮箱和验证码的校验。如果成功，存储新注册的用户信息。反之，返回失败信息给用户。负责注册的安卓活动类为RegisterActivity。RegisterActivity主要有两个监听事件，一个就是获取验证码监听，一个为注册按钮监听。如果用户注册成功，会回显用户，并且跳转到登陆页面。

图4.8  用户注册流程图

**4.5.1.2 用户登陆流程**

    大概流程如图4.9，用户进入登陆页面，输入邮箱密码信息，然后点击登陆按钮。后端UserService收到登陆请求信息后，首先进行参数的校验。如果参数校验成功，调用UserDao，根据邮箱去查询用户的账户信息。如果查询到用户信息，UserService对密码进行比对校验，如果密码一致则登陆成功。否则登陆失败。如果未查询到用户信息，则提示该账户不存在。如果不密码不正确，提示用户密码不正确文案。

    用户登录安卓活动类为LogonActivity。该类有两个监听事件，分别为登陆和注册事件监听。如果是注册事件，则跳到注册页面。如果是登陆则会进行参数校验，然后调用后台登陆接口。如果成功，则会将用户id信息存储到缓存中，然后跳入APP主页面。

图4.9  用户登录流程图

**4.5.2 作业检查模块设计**

    作业检查模块功能是获取练习作业、检查练习作业、检查照片作业。

**4.5.2.1 软件单元构成**

    作业检查模块对应的类图如图4.10,其中WorkManagerController主要是API层，为客户端提供HTTP接口。CheckServiceImpl主要提供作业模块的业务实现，该类实现了CheckService接口。

    CheckService依赖四个类去完成逻辑实现。ErrorContinerService为错题管理类。也就是一个错题容器，存储了用户的所有错题。CheckHistoryService为检查历史接口。ErrorHistoryService为错题历史接口。EvalVisitorImpl为四则运算表达式检查实现类，其依赖CalVisitor进行表达式检查。CalVisitor继承自CalExprBaseVisitor。

图4.10  作业检查模块类图

**4.5.2.2 拍照检查流程**

    用户点击拍照检查按钮，可以从相册或者通过调起相机拍摄照片获取作业图片。主要实现在CheckFragment中。最后会调用后台上传图片检查作业的接口。

    后台流程如图4.11，接收到客户端的图片后，先会存储到磁盘，然后通过WebOcr去进行图片识别。如果识别失败直接返回。成功则会返回到CheckService。CheckService根据结果进行计算。

    计算流程如下：

1. 根据算术表达式获取抽象语法书。
2. 实例化计算逻辑类。
3. 根据抽象语法树和类进行运算。

    计算成功后，会记录检查历史，并将错题记录到错题历史。然后返回结果id。客户端接收到id后调用对应接口获取结果数据。并渲染到客户端。客户端采用ListView布局进行渲染。

图4.11  拍照检查流程图

**4.5.2.3 获取练习作业流程**

    用户输入练习数量，点击获取按钮。主要实现在PracticeFragment中。这个Fragment中有一个button监听。获取成功之后，进入PracticesActivity。PracticesActivity通过ListView去渲染页面。

    后台流程图如4.12，接收到客户端请求后，先进行参数校验，然后根据用户id去Redis获取用户的错题。这里通过Set去存储用户的所有错题。每次根据数量pop出来。如果数量不足，需要去UserService去获取用户信息。根据用户的年级去生成对应的练习题。这里将生成过程解藕出来，便于后期的功能实现。生成成功后，直接返回结果。

图4.12  获取练习作业流程图

**4.5.2.4 检查练习作业流程**

    检查练习作业和检查图片作业流程一致，如果跳过图片上传过程，就和检查图片作业流程完全一致。

**4.5.3 作业历史模块设计**

    作业历史模块功能是管理图片检查和练习检查历史以及所有检查中的错题历史。

**4.5.3.1 软件单元构成**

    作业历史模块对应的类图如图4.13,其中WorkHistoryController主要是API层，为客户端提供HTTP接口。CheckHistoryService主要提供作业检查历史的业务实现。ErrorHistoryService提供作业错误历史的业务实现。历史的数据来自于每次检查过程。所以这两个Service对于API层只需要提供对应的查询接口即可。

图4.13  作业历史模块类图

**4.5.3.2 获取历史流程**

    获取历史分为练习历史、拍照检查历史、错题历史，流程基本一致。下面我们参考获取练习历史的流程。

    用户点击获取练习历史的按钮。主要实现在CheckHistoryFragment中。页面初始化的时候直接调用后台接口获取历史数据。返回之后依赖ListView和Adapter渲染到页面。

    后台流程很简单，直接从数据库根据条件查询对应数据返回即可。

**4.5.4 班级模块设计**

    班级模块主要功能是班级管理、家长加入班级、班级用户管理。

**4.5.4.1 软件单元构成**

    班级模块对应的类图如图4.14，ClassController为班级模块的API层。ClassServiceImpl提供该模块的业务逻辑实现。这个类依赖了UserService和ClassWorkService。依赖UserService获取用户信息。依赖ClassWorkService获取班级作业信息。ClassDao为班级的持久化层，用来存储所有存在的班级。ClassUserDao为班级和用户的映射表，存储所有班级对应的用户信息。

图4.14  班级模块类图

**4.5.4.2 创建班级流程**

    老师点击创建班级按钮，进入创建班级页面，页面活动实现在TeacherActivity中。这个页面有三个按钮监听：返回监听、选择学校、创建。点击选择学校则会跳入选择学校页面，页面活动实现在SearchSchoolActivity中。获取学校逻辑比较简单，根据用户输入模糊查询所有学校即可。创建按钮主要就是调用后台接口进行班级创建。

    后台创建班级流程如图4.15，接收到创建请求后先去查询是否存在当前班级，如果存在直接返回。将班级信息存储到数据库。根据ID去生成班级ID，将班级ID更新到班级信息，返回结果。这里依赖数据库去生成班级唯一ID。但是会带来double IO的性能问题。后期可以借助Redis实现分布式班级ID生成。

图4.15  创建班级流程图

**4.5.4.3 加入班级流程**

    家长点击加入班级，进入页面，页面活动实现在ParentActivity中。该页面除了返回按钮，还有一个加入按钮监听。加入按钮监听会调用后台加入班级页面。成功之后结束页面活动。

    后台加入班级流程如图4.16，接收到加入请求，后台根据班级编号去查询班级是否存在。如果不存在直接返回提示用户。否则构建用户和班级的映射信息，将信息持久化到后台数据库，返回成功，

图4.16  加入班级流程图

**4.5.4.4 解散班级流程**

    解散班级活动在MyClassActivity中实现，页面初始化的时候会获取所有班级信息。并且为每个班级设置按钮监听。老师点击解散班级，客户端会直接调用后台解散班级接口。成功后移除对应班级。如果班级中有成员信息，需要老师先移除所有成员。

    后台解散班级流程如图4.16，接收到请求后，判断班级是否存在成员，如果存在返回提示用户需要移除所有成员。否则判断是否存在该班级。如果已删除，直接返回。如果班级存在，删除所有班级作业信息。最后删除该班级信息。

图4.16  解散班级流程图

**4.5.4.5 退出班级流程**

    退出班级按钮也在MyClassActivity中实现。家长点击退出按钮后调用后台，成功后移除该班级信息。

    后台接收到退出请求，调用持久化层删除用户对应的该班级信息，并且移除该班级所有的作业信息。返回即可。如果班级不存在直接返回失败。

**4.5.4.6 移除班级用户流程**

    移除用户活动在MyCLassStudentActivity中实现。在老师点击获取用户或者老师解散班级但班级存在用户的时候被创建。该页面初始化的时调用后台获取班级所有成员信息。并且为每个用户设置移除监听。点击移除用户，会调用后台移除用户接口。如果成功，将对应行移除掉即可。

    后台接收到移除用户请求，调用持久化层删除对应用户和班级的映射信息，返回。

**4.5.5 班级作业模块设计**

    班级作业模块主要功能是作业的管理。

**4.5.5.1 软件单元构成**

    班级作业模块对应的类图如图4.17,其中ClassWorkController主要是API层，为客户端提供HTTP接口。ClassWorkService主要提供班级作业历史的业务实现。ClassWorkDao用来持久化作业信息。ClassService用来获取用户所有的班级信息。

图4.18  班级作业模块类图

**4.5.5.2 获取用户所有的班级作业**

    获取用户所有班级作业在CLassFragment中实现。通过ListView去渲染作业列表。

    后端流程如图4.19，接收到客户端请求，首先获取用户加入的所有班级。然后根据班级信息去获取所有作业信息。最后组装获取到的所有信息，返回给客户端。如果没有获取到班级信息或者作业信息，直接返回空。

图4.19  班级作业模块类图

**第五章 系统开发**

**5.1 开发环境**

    开发工具：Android Studio、Android SDK、JDK、IDEA

    开发环境：Mac 13（开发）、魅族手机、Google安卓虚拟机

**5.2 关键技术**

**5.2.1 客户端服务器通信框架**

     该项目的客户端通信协议使用传统的HTTP。因为依赖HTTP通信，所以在应用层使用SpringMVC做了请求的转发工作。对于安卓应用的HTTP通信，大多使用的为okhttp框架。为了清楚实现流程，更加熟悉的了解项目原理，下面会分析介绍okhttp。架构如图5.1所示：

图5.1  okhttp架构图

    所谓okhttp，底层当然是HTTP协议。其流程如下：

1. OkHttpClient是通过Builder模式去创建的，通过Builder模式，我们可以更加容易的创建对象，保证了对象的一致性。创建成功后，会返回一个OkHttpClient对象。拥有了这个Client，我们就可以和服务器进行HTTP通信了。对应每一个连接通信，我们都是有一定的数据结构和对应的序列化协议。在该项目中，大部分使用JSON进行对象的序列化和反序列化。数据采用JavaBean的方式去构建。
2. 当我们需要发送一条HTTP请求的时候，我们首先应该构造一个Request对象。首先我们拿到对应的客户端数据。如果是表单的方式，我们直接可以通过RequestBody构建一个表单的Request，传入对应的url即可。如果是JSON格式，我们首先创建一个JavaBean对象，然后封装对应数据，最后使用JSON进行序列化并构建Request对象。
3. 创建Request成功后，我们只需要调用他的newcall方法。会返回一个RealCall。我们可以直接调用去Execute方法执行对应请求。或者我们调用enqueue方法，将其加入到队列。然后通过Dispatcher去分发执行。当然后者的效率更高。因为其通过队列异步了网络IO。
4. 后续的流程都是框架内不实现。通过拦截器对请求进行拦截分析。根据不同的逻辑实现不同的操作。比如执行缓存，或者判断是否降级的。最后一个拦截器就是获取对应的连接执行真正的IO操作。
5. 成功后，通过执行我们的回调方法，我们只需要根据请求的响应逻辑进行对应的实现即可。

    对于后台，使用Tomcat服务器，通过IO多路复用模型处理对应的网络请求。Tomcat基于HTTP，去解析对应的请求。将数据转发到对应的Servlet。因为我们使用SpringMVC框架。所以通过全局的DispatcherServlet去拦截所有请求，根据业务实现分发到不同的Handler去处理，最后返回。

**5.2.2 服务端SpringMVC架构**

**5.2.2.1 SpringMVC架构流程**

    后端项目使用SpringMVC完成项目的搭建，对于SpringMVC的实现，也进行了必要的研究。其架构图如图5.2所示。

图5.2  SpringMVC架构图

    SpringMVC依赖Tomcat服务器去处理HTTP请求。该框架实现了一个DispatcherServlet，也就是一个全局的拦截器。Tomcat的所有请求都是通过该Servlet去处理。DispatcherServlet收到请求后会调用HandlerMapping处理器映射器。该映射器主要是帮助DispatcherServlet去查找对应的处理器。他会根据请求的uri进行查找。如果找到，通过HanlderAdapter去调用处理器执行。执行完成后，对应处理器会返回ModelAndView。HanlderAdapter会将返回的ModelAndView返回给DispatcherServlet，DispatcherServlet会进行对应操作后返回给请求方数据。

    通过上面的流程，我们可以了解到其全局依赖DispatcherServlet。并且有一个全局的异常处理。在该项目开发的过程中，经常会遇到网络调用的问题，都是通过在DispatcherServlet打断点调试解决问题。

**5.2.2.2 SpringMVC在项目中的实现**

    下面根据用户登陆去描述其实现。

1. 客户端调用HTTP请求，uri为/user/login。
2. 服务端Tomcat接收到http 请求后解析，然后交给DispatcherServlet
3. DispatcherServlet调用HandlerMapping。HandlerMapping通过/user/login去查找对应的处理器返回。
4. DispatcherServlet通过HanlderAdapter去调用处理器UserController中的login方法。
5. 后端执行对应的login逻辑后返回。
6. DispatcherServlet收到数据，进行处理并返回给Tomcat。
7. Tomcat构建Response返回给调用方。

**5.2.3 Redis数据存储**

**5.2.3.1 Redis架构原理**

    Redis是一个传统的内存数据库，它和关系型数据库Mysql的不同之处在于：它是将数据存储到内存中，通过KV的形式存储。它的优点就是访问速度快。为了更好的将Redis使用到项目中，需要了解Redis的架构实现以及其存储数据的数据结构。

    Redis架构图如图5.3：

图5.3  redis架构图

    在项目中，通过Spring-Data-Redis去操作Redis，但是底层还是通过Jedis去进行连接通信的。归根结底，最终还是通过Socket建立TCP连接。如图5.3，Redis服务器通过IO多路复用去处理所有客户端连接。每当有时间到达，他会将请求事件转发给文件时间分派器。分派器根据时间类型去选择对应的事件处理器。如果是第一次建立连接，那么他会交给连接应答处理器去处理连接请求。如果是命令，则是由命令处理器处理。

    我们的一次添加操作，在Jedis则会转化成对应命令数据。数据协议采用Redis内部实现的序列化协议。Redis服务器接收到请求数据后，会根据对应的协议进行反序列化，得到后会进行语句分析，如果语句错误，会返回异常。所以在项目开发中，如果我们理解这个过程，遇到问题也能非常容易的去定位。如果语句正确，Redis则会根据内部流程执行。

    我们需要注意的是Redis对应命令执行过程中采用的是单进程的方式。何为单进程，其实就是我们不需要考虑多进程并发的情况。因为同一个进程在同一时间只会在一个CPU上执行。

**5.2.3.2 Redis数据结构**

    Redis内部实现了五种数据结构，分别为字符串，哈希表，链表，集合，有序集合。

* 字符串：可以存储一个字符串
* 哈希表：可以存储一个Map，通过键值对映射
* 链表：和传统的链表一样，每个值都是通过指针进行连接，可执行push和pop操作，数据有序
* 集合：可以存储多个字符串元素，数据是无序的。并且是不可重复的。支持结合间的操作
* 有序集合：和集合不同的就是，数据是有序的。根据设置的分值进行排序，底层通过跳跃表实现。访问速度为对数级别。

    项目中注册验证码、以及玩家的错题数据都是通过Redis去存储的。验证码使用字符串的数据结构进行存储。玩家错题采用集合的数据结构。对于重复错题，我们通过set，保证不会重复存储。每一个用户拥有一个set集合，传统意义上我们可以称其为用户的错题池。错题在练习中也会被使用到。因为在用户获取练习的时候，优先选择用户的错题数据。我们通过集合的一些操作，可以随机pop出指定数量的错题，效率是非常可观。

**5.4 开发成功**

用户注册页面如图5.4,用户输入对应信息即可进行注册。

图5.4  注册页面

用户登陆页面如图5.5，用户输入注册成功的信息进行登陆。

图5.5  登陆页面

检查结果页面如图5.6，用户上传作业照片后会返回该页面。页面为检查结果。

图5.6  检查结果页面

拍照页面如图5.7，用户可以通过相册或者拍照的方式进行图片检测。

图5.7  拍照页面

练习结果页面如图5.8，用户输入练习题目数量，点击练习按钮，会进入该页面。

图5.8  练习结果页面

个人信息页面如图5.9，点击昵称后可进入个人信息页面。昵称和年级可修改。

图5.9  个人信息页面

数据历史页面如图5.10，点击错题本或者学习历史可进入该页面。

 图5.10  数据历史页面

创建页面如图5.11，点击创建班级进入该页面。

 图5.11  创建班级页面

 我的班级页面如图5.12，该页面展示用户加入的所有班级信息。

 图5.12  我的班级页面

 发布作业页面如图5.13。

 图5.13   发布作业页面

 作业列表页面如图5.14。

 图5.14   作业列表页面

**第六章 系统测试**

**6.1 测试设计**

**6.1.1 测试环境**

    阿里云CentOS，用于运行后台服务。

    安卓手机，用于运行客户端程序。

**6.1.2 测试范围**

    测试项目主要分为三类。第一为拍照识别、练习、历史功能测试。第二位用户模块（注册、登陆、用户信息）。第三位班级模块功能测试。详细数据如下表。

    表6.1 拍照识别、练习、历史功能测试

|  |  |
| --- | --- |
| 测试项 | 测试目标 |
| 拍照识别 | 确认可上传图片，并能正确返回识别结果 |
| 练习获取识别 | 确认可获取联系数据，并能正确返回判别结果 |
| 历史数据展示 | 确认可展示数据，并且数据正确 |

    表6.2 用户模块功能测试

|  |  |
| --- | --- |
| 测试项 | 测试目标 |
| 用户注册 | 确认可注册，一个邮箱只能注册一次 |
| 用户登陆 | 确认存在的账号可登录 |
| 用户数据更新 | 头像、昵称等信息可修改 |

    表6.3 班级模块功能测试

|  |  |
| --- | --- |
| 测试项 | 测试目标 |
| 班级创建 | 老师可创建班级，同一用户的同一班级只能被创建一次 |
| 班级加入 | 家长根据存在的班级号可加入班级 |
| 班级数据展示 | 家长和老师班级数据展示正确（成员信息、班级号等） |
| 作业发布 | 老师可发布作业 |
| 作业数据展示 | 老师和家长作业数据展示正确 |
| 班级管理 | 老师可成功解散班级，存在班级成员不允许解散 |
| 班级成员管理 | 老师可以移除已创建班级的成员，家长可以退出班级 |

**6.2 测试用例及测试记录**

    表6.4 测试记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项 | 步骤 | 输入 | 预期结果 | 实际结果 |
| 拍照识别 | 通过系统相册拍照或者从相册获取作业图片，点击确定 | 作业照片数据 | 返回正确的识别结果 | 与预期一致 |
| 练习获取识别 | 点击获取练习题目，输入答案，提交 | 点击练习，输入练习答案 | 返回正确的判别结果 | 正确返回 |
| 历史数据展示 | 分别点击检查、练习、错题历史按钮 | 点击按钮 | 按日期返回正确的检查历史 | 正确返回 |
| 用户注册 | 点击注册按钮，输入正确信息，点击获取验证码，查看验证码输入。点击注册按钮 | 注册信息、注册验证码、点击注册按钮 | 注册成功，跳转到登陆页面 | 与预期一致 |
| 用户登陆 | 输入用户信息，点击登陆按钮 | 账号信息，点击登陆按钮 | 登陆成功，跳入APP主页面 | 成功登陆，跳转正确 |
| 班级创建 | 点击创建班级按钮，填写班级信息，点击创建 | 班级信息、点击创建按钮 | 创建成功，返回班级列表 | 与预期一致 |
| 班级加入 | 点击加入班级，输入班级号，点击加入 | 班级号 | 加入成功，返回班级列表 | 与预期一致 |
| 班级数据展示 | 点击我的班级 | 点击我的班级按钮 | 返回我的班级列表 | 返回数据成功，数据正确 |
| 作业发布、作业数据展示 | 点击发布作业，输入作业信息，点击发布 | 作业信息 | 发布成功，返回作业列表 | 发布成功，列表数据显示正确 |
| 班级管理 | 点击解散班级 | 点击解散班级按钮 | 如果班级成员为0，解散成功，否则提示用户并跳入移除成员页面 | 与预期一致 |
| 班级成员管理 | 点击退出班级  选择班级成员，点击移除 | 点击按钮 | 退出班级成功  移除成员成功 | 与预期一致 |

**6.3 测试结果及结论**

**6.3.1 测试用例执行结果**

    所有测试用例均测试通过，并且实际结果和预期结果一致。

**6.3.2 软件缺陷分析**

    项目采用安卓原生开发，但是由于自己刚接触安卓知识，因为在很多需求实现过程中并不是很人性化，只是将功能实现。项目部分需求在逻辑上会有一些漏洞，思路不够严谨，导致用户体验并不是很佳。安卓端的异常处理并不是很完善，在遇到漏洞可能会导致项目闪退。

**6.3.3 测试结论**

    项目已通过所有功能测试，需求已全部实现。后续定制更加完善的压测计划，在压测中发现和解决问题。

**第七章 总结和展望**

**7.1 本文工作总结**

    本文基于安卓以及后端技术，实现了智能教辅系统。践行了传统移动互联网的研发模式。项目创新之处在于对于新技术的实践应用。让移动产片更加智能化。

    项目虽然已经完全可用，但是在敏捷开发模式下，使得项目还是存在设计上的不足。对于项目的研发，最终还是要落地到用户中去。因为需要对项目发版并进行灰度测试，才会体现出真正的价值。

**7.1 未来工作展望**

    该项目只实现了简单的四则运算图像识别，并且没有将结果展示到原图上。后续需要对这方面进行改进。主要为下面两方面：

1. 通过大数据和人工智能技术，实现应用数学的智能化判断。
2. 将识别结果可视化到原图上。这需要对图像识别进一步探索。

    项目开发采用敏捷开发模式，初期功能相对简单，后期会增加项目需求。主往家长和老师互动方面靠近。让家长了解孩子的更多学习情况，充分发挥家长的价值。

    项目部分UI需要进行优化。通过深入学习安卓布局实现。

**结束语**

    本文从项目背景、项目需求、系统研发、技术原理、项目成果、项目测试多方面切入，详细介绍了教辅APP的系统设计以及实现。项目价值意义较大，依赖移动互联网以及机器学习服务于教育行业。对互联网教育的探索更近一步，为今后的教育发展指明方向。

    在项目开发过程中自己成长非常大，养成了独当一面的能力，能快速定位问题，解决问题。后续还需继续维护更新项目，加入更多的功能，让应用可以真正落地到教育行业内部。