

华中科技大学

《软件工程》项目报告

题目： 不刷题小程序

课程名称： 软件工程

专业班级： CS2009

组 名： XHZB

同组成员：学号： U2020xxxxx

姓名： xym

学号： U2020xxxxx

姓名： hjl

学号： U2020xxxxx

姓名： zrk

学号： U2020xxxxx

姓名： bkl

指导教师： 瞿彬彬

报告日期： 2022.11.8

计算机科学与技术学院

任务书

一 总体要求

1. 综合运用软件工程的思想，协同完成一个软件项目的开发，掌握软件工程相关的技术和方法；
2. 组成小组进行选题，通过调研完成项目的需求分析，并详细说明小组成员的分工、项目的时间管理等方面。
3. 根据需求分析进行总体设计、详细设计、编码与测试等。

二 基本内容

根据给出的题目任选一题，自行组队，设计与开发中软件过程必须包括：

1. **问题概述、需求分析：**正确使用相关工具和方法说明所开发软件的问题定义和需求分析，比如 NABCD 模型，Microsoft Visio，StarUML 等工具 (20%)；
2. **原型系统设计、概要设计、详细设计：**主要说明所开发软件的架构、数据结构及主要算法设计，比如墨刀等工具 (35%)；
3. **编码与测试：**编码规范，运用码云等平台进行版本管理，设计测试计划和测试用例 (30%)；
4. **功能创新：**与众不同、特别吸引用户的创新 (10%)；
5. **用户反馈：**包括用户的使用记录，照片，视频等 (5%)。

目 录

任务书.....	I
1 项目介绍.....	1
1.1 项目背景及意义.....	1
1.2 项目基本目标.....	4
1.3 可行性分析.....	4
1.4 人员管理和项目进度管理.....	5
2 需求分析.....	8
2.1 需求分析概述.....	8
2.2 UML 相关需求分析图.....	8
2.3 原型系统设计.....	9
3 概要设计和详细设计.....	14
3.1 系统结构.....	14
3.2 类图.....	16
3.3 关键数据结构定义.....	17
3.4 关键算法设计.....	21
3.5 数据管理说明.....	22
3.6 界面设计.....	25
4 实现与测试.....	33
4.1 实现环境与代码管理.....	33
4.2 关键函数说明.....	36
4.3 测试计划和测试用例.....	37
4.4 结果分析.....	40
5 总结.....	41
5.1 用户反馈.....	41
5.2 总结.....	44
5.3 改进方向.....	45
6 体会.....	46

1 项目介绍

1.1 项目背景与意义

1.1.1 项目需求 Need

1. 项目来源：儿童学习

小明的妈妈近期很郁闷，小明上小学后，老师要求每天完成 30 道口算题，每天出题改题对小明妈妈成了一个大负担，“有个可以出题的软件就好了”小明妈妈这么想。并且小明现在处于一年级，小明妈妈希望买一个软件可以用三年。

目前许多的家长都面临这样的问题：老师所布置的作业往往需要孩子和家长共同完成，孩子写完作业需要家长批改或者家长全程参与。然而许多家长由于工作繁忙或者专业知识有限并不能好好的完成该项任务，但是孩子需要一个可以提供题目和批改功能的东西来帮助自己完成老师布置的作业，因此，对于广大的家长而言，迫切需要一个可以提供题库和自动批改题目的小程序来帮助自己更高效更准确的完成任务。

2. 多重需求调研分析

在当下的社会，对于从小学到高中，乃至大学或者工作的人群而言，许多人都需要“做题”。小学生、初中生、高中生需要通过做题来帮助自己巩固知识和测试；大学生也需要通过做题来提高自己的学习效率和检测成果；想要找工作的人也需要通过做题来更加熟悉面试题目或者笔试题目。

对于大家而言，时间是一个很宝贵的东西，我们在往常会通过购买书籍的方式来做题，然而这会需要我们自己订正答案，这是一件费时费力的事情。同时对于做题的人而言，重要的往往是错题，做题者往往想要将做题集中放置在一起并将错误结果和正确结果展示给自己。因此对于所有需要做题的人而言，会产生一个可以提供题目、题目自动批改，将错题呈现给自己的可以随时随地使用的小程序或者软件的需求。

此外，当下人们所拥有的碎片化时间越来越多，往往不能花很大块的时间坐在“自习室”刷题，因此需要一个可以随时随地拿出来做一道题的工具来帮助自己实现知识的测试和巩固。

综上所述，大家的广泛需求就是一个可以提供题库、可以自动批改、可以收

集错题、随时随地可以拿出来使用的方便的做题工具。

1.1.2 实现方法 Approach

展现形式上，采用微信小程序的方式向用户开放该做题工具，将其发布到线上，使得用户可以通过扫描二维码或者微信搜索的方式查询到该小程序。

实现形式上，采用了类似于 MOOC 答题和批改的形式展现题目给读者，并且提供了单独的错题集、收藏夹等。

开发途径上，采用了微信小程序官方开发工具，利用 wxml、wxss 对小程序界面进行多层渲染，利用 json 对整体的界面进行配置，利用 js 实现界面功能和后端；同时使用小程序整体的 app 文件对整个小程序之间的连接进行配置；采用微信开发者提供的数据库保存用户数据和题库数据，同时在 js 中提供接口将用户的小程序和数据库连接起来。同时采用在线文件转换工具将所收集到的题集转换成数据库所能读取的格式统一的保存在数据库中。

1.1.3 项目益处 Benefit

对于广大需要从孩子教育中解放时间的家长而言，他们可以通过该小程序为孩子提供更加有针对性的题目以及更加准确的答案和更加高效的批改效率。不仅可以节省自己在该任务的时间，而且还可以查看到孩子的做题情况和错题集，来更好的帮助孩子分析目前孩子所面临的问题，以此来帮助孩子制定下一步的培养计划。同时小程序提供了学习报告等功能，可以在家长将手机给予孩子做题的时候帮助家长监督孩子目前的学习状态。

对于需要做题的人而言，他们可以随时随地的打开小程序打开自己正在刷的题集并做题，不管是在碎片化时间中需要在琐碎的时间做一道题还是在整个大块的时间需要做一整个题集，都可以为其提供该功能。同时提供收藏功能将做题者目前不确定的题目加入收藏集方便后续查看，提供自动加入错题集的功能并且按照不同的题集对其进行了分类，方便用户查看。利用该小程序，做题者可以更加高效的做题。

1.1.4 竞争对手 Competitors

各类的做题软件和做题小程序都是我们的竞争对手。目前市面上有许多的做题软件或者做题小程序，比如“刷题神器”、“我爱刷题”等等，其中“刷题神器”提供了很丰富的功能。

相比竞争对手，我们具有几点优势：

①简洁性：很多的刷题软件和小程序都追求太多的功能导致界面乱而杂，使得用户点进去的时候眼花缭乱，并不太好找到自己所需要的东西。我们的小程序将用户所需要选择的题库入口、正在刷题的入口、收藏夹入口直接放在首页使得用户可以直接进入做题。

②更加方便的入口：我们专门设置了一个正在刷题的入口，该正在刷题进入的是用户选择的当前想要刷题的题库。该想法参考了《不背单词 APP》的背单词功能，不背单词每次在首页提供背单词的入口，用户选择一本词书开始背之后便可以通过该入口直接进入该词书进行单词背诵。该功能考虑用户在一段时间内想要专注刷一本题集，因此提供给用户一个更加方便进入刷题的入口。

③自动生成错题集：用户在做题直接提交答案会自动批改，在提交后，与正确答案不符的题目将直接加入到错题集中，错题集按照科目分类，用户可以随时随地查看错题集并进行重做、删除等操作来巩固知识。

④自动生成做题报告：在做完一本题集后，会为用户提供整个题集的做题情况，该想法参考了 MOOC 的做题功能，将所有题目的对错情况呈现给用户，并且可以点击题目具体查看一道题目当时的做题情况和正确答案。

⑤生成学习报告：用户可以知道自己做了多少道题，错了多少道题和正确率，使得用户更有成就感，提高做题的热情。

1.1.5 推广途径 Delivery

1. 官方发布：

生成二维码方便用户扫描使用，同时在官方平台上线，在微信搜索框中直接搜索“不刷题”就可以看到我们的小程序。

2. 转发推广：

通过 QQ、微信、微博等平台对该小程序进行推广。

3. 用户反馈：

一个优质用户可以帮助软件带来数十个潜在用户，因此可以采用奖励机制让用户对小程序进行评价和推广，一传十十传百来通过用户自主传播的方式使得小程序得到推广。

4. 广告推广：

可以通过公开平台比如微博、b 站、小红书、今日头条等进行小程序介绍，并将其打上标签，方便软件将其分类并推送给合适的人群。

5. 名人效应：

可以利用小程序做出比较有意思的视频、动图、图片等，如果这些形式的东西可以取得火热的播放或者传播，会有更多的人知道小程序，从而得到更多的用户。

1.2 项目基本目标

“不刷题”作为一款针对小学生的刷题小程序，意在使所有小学生、小学教师、小学生的家长都能自由地刷题、出题、监督学习，使小学生随时随地拿起手机就能通过刷题学习，使老师能够通过海量题库收获出题的灵感，使家长能够通过这个工具检测、督促孩子的学习。

为了实现以上目标，我们在功能实现上做了如下设计：

- 1、小程序整体风格简约清晰，没有冗杂的内容，不会分散孩子的注意力，使孩子专注于学习。
- 2、题库丰富，拥有小学一到六年级的语文、数学、英语三个科目的海量题目，后台管理人员还会定期更新题库，保证小程序的实时性与新颖性。
- 3、刷题功能齐全，包含查看答题结果、收藏题目、记录错题、删除收藏和错题、自动生成学习记报告的功能。
- 4、设计合理，刷题、收藏、错题集有快捷入口，可以快速进入，方便省时，尤其是正在刷题快捷入口的实现，可以让用户像选择一本单词书一样选择一个感兴趣的题集添加到我的题库，点击正在刷题即可进入当前题集刷题，非常方便，避免用户每次想进入该题集时都要在题库中寻找，可谓是本小程序的一大特点和亮点。

1.3 可行性分析

为了方便老师家长和学生的使用，我们使用微信小程序来实现不刷题这个软件。从技术的角度来看，我们需要实现获取题库、用户信息和题目的保存和界面的实现。在获取题库方面，现在网上有很多不同年级不同科目的题目供我们使用，我们可以把题目下载下来，然后通过插件把题目的内容变成 json 文档来使用。在用户信息和题目保存方面，我们使用微信的云开发来对数据进行分类保存。在

界面的实现方面，我们可以根据墨刀上的原型设计来实现我们不刷题小程序的不同界面。

在确认可实现性后，我们使用 COCOMO 模型对工作量进行定量评估。记以人月（PM）为单位的工作量为 E ，开发时间为 D 月，项目的代码行数估计值为 L （千行）。对于该项目的组织型软件，有模型公式如下：

$$\begin{cases} E = 2.4 \times L^{1.05} \\ D = 2.5 \times E^{0.38} \end{cases}$$

经小组的讨论与综合评估，取 $L = 5$ ，解得 $E \approx 13.01$ ， $D \approx 6.63$ 。所以具体参与软件开发的人数 $N = E/D \approx 2$ 。值得注意的是，上面计算出的 E 和 D 只作为中间结果，要对工作量进行更准确的评估需要采用中间 COCOMO 模型，加入 EAF 调节因子（使模型考虑到我们项目的特殊性和简易性）。基于这是一个课程项目而非商业项目的事实，对各个调节因子取值得到 $EAF = \prod_{i=1}^{15} F_i \approx 0.26$ 。这样可得 $E' = E \times EAF \approx 1.95$ ，即需要接近两个月的开发时间，可以在软件工程结课验收前完成。所以，该项目具有很高的技术可行性。

从市场的角度来看，正如需求分析中所述，当令许多家长工作繁忙不能过多时间监督孩子做题，同时小孩也需要很多的题目来巩固他们所学的知识，所以我们不刷题小程序有着很大的市场空间。

从经济的角度来看，小组成员使用各自的计算机设备即可完成主要的开发工作；利用微信小程序的开发平台和 Gitee 线上平台，团队可以方便地以线上线下结合的方式开展交流讨论以协同工作进度。所以项目的初步开发不存在大额的成本，具备经济上的可行性。

1.4 人员管理和项目进度管理

1.4.1 人员管理

基于项目团队的成员情况，采用开放与同步相结合的人员管理组织范型。首先基于基本目标对项目进行自然划分，之后通过民主的形式交流讨论，结合各人擅长领域进行初次职责分配。另外由于项目人员较少，交流更加及时方便，在项目的进行上充分发挥了各自的创造性，及时动态更新补充方案。四位小组成员的具体分工如下：

xym 负责项目全过程的文档整理，分支项目的集合，总体把握项目进度。负责项目后端开发，主要包括数据库交互，算法设计。参与小程序主体设计与软

件测试。

hj1 负责项目执行阶段成员的进度统计,UML 相关需求分析和甘特图绘制。负责数据库的设计,管理以及部分小程序后端开发,负责软件的上传与发布。

bkl 负责软件需求分析,基于墨刀平台进行 UI 设计,并负责前端开发,以及界面优化以及 logo 设计。

zrk 负责软件 UI 设计与软件核心交互部分的开发,负责软件测试,参与小程序主体设计与软件测试,负责用户反馈的收集与整理。

1.4.2 项目进度管理

总体上,通过甘特图完成项目初期具体任务的分派与规划;通过燃尽图在项目的全过程中对进度进行跟进和分析。

项目总体任务划分为:Gitee 平台项目立项,UI 设计,软件设计,软件测试,讲解与展示五个部分。将这 5 个阶段进一步细化成了一个个单独可执行的任务并进行了分配,项目甘特图如图 1.4.1 所示。

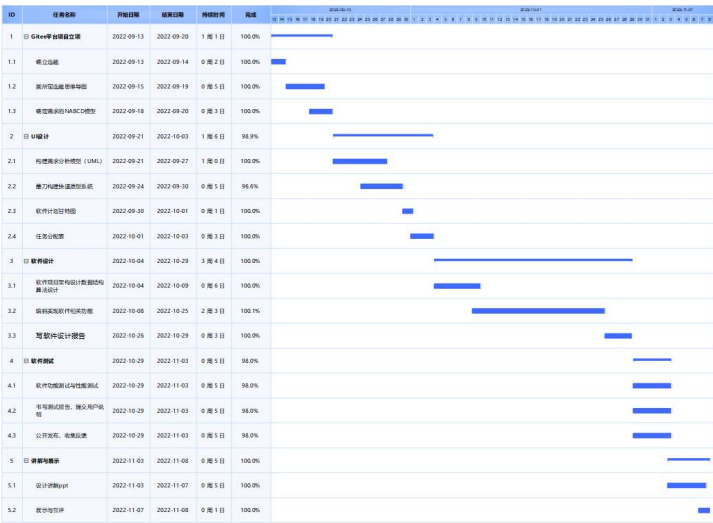


图 1.4.1 项目甘特图

小组各成员按照甘特图中所分派的具体任务完成项目,过程中定期进行进度统计,并通过燃尽图来判断当前进度是超前还是落后于预期,小组进度燃尽图如图 1.4.2 所示。由燃尽图可以看出在 10 月初期进度稍有落后,随机加快了进度。

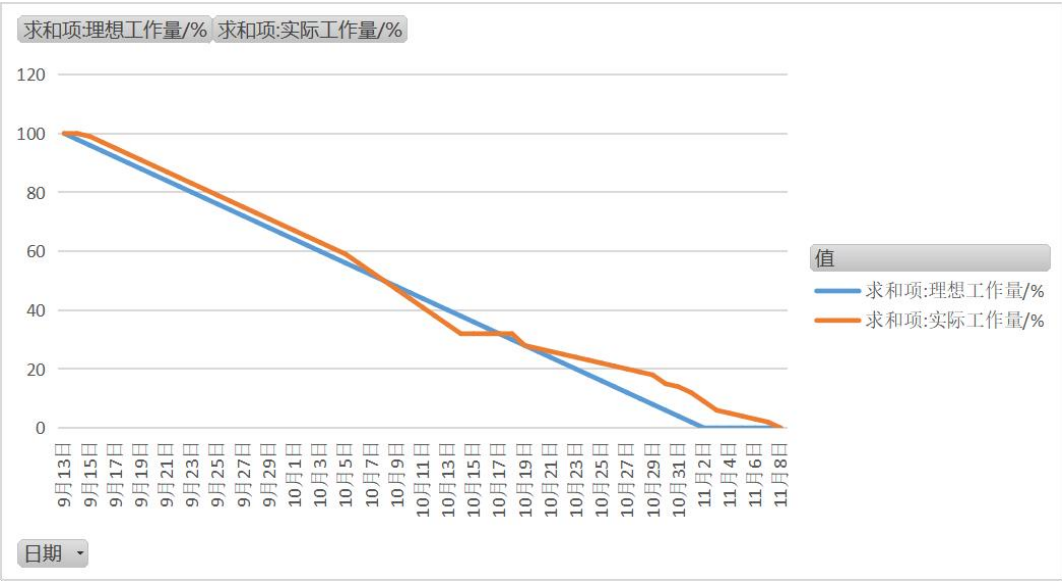


图 1.4.2 小组进度燃尽图

2 需求分析

2.1 需求分析概述

需求分析过程首先从用户需求方面思考。教育问题一直是社会的一大重点，而小学可以说是教育的起点，家长们都非常重视，但课堂上所学的内容有限，也不一定适合所有人，许多孩子被要求去上课外补习班来巩固、拓展所学知识，或者买了一大堆教辅资料刷题，但无论是去补习班上课还是做教辅资料，都需要一定的时间和金钱成本，有些家长工作比较忙，或者经济条件不佳，没有条件送孩子去课外补习班学习，而且教辅资料大多比较厚重，难以携带，往往需要在固定的时间地点才能刷题，灵活性不高，所以我们设计了一款微信小程序，只要一部手机就能实现随时随地轻松刷题，无论学生身处何地，都可以随时打开小程序，开启刷题之旅，这样能极大减轻家长的负担、提高效率、降低学习成本，这就是我们选择设计一款刷题小程序的原因。

满足了用户随时随地无负担刷题的需求，还要满足用户在刷题过程中的需求。比如做错的题目能记录在一起，方便复习巩固，就像现实中的错题本一样，所以我们的程序需要错题本功能，在用户刷题的过程中自动记录用户的错题；当用户遇到认为比较经典值得反复刷的题目时，也会希望这些题被记录在一起，所以我们的程序还需要收藏功能，在用户刷题过程中，点击收藏即可收藏改题目，可以前往我的收藏页面查看收藏的题目；用户为了节省时间提高效率，需要快速进入自己心仪的题集、快速查看错题、快速查看收藏的题目，所以我们的程序需要设计实现一些快捷入口，点击即可进入刷题或查看；刷完题后，用户可能会希望查看自己的学习时长、刷题数，分享自己的学习成功，增加学习的仪式感和成就感，所以我们的程序需要有自动生成学习报告功能，可以查看今天和累计的刷题数和学习时长，并分享；最重要的是，每个用户的错题、收藏、阶段、心仪题集不一样，登录后获取用户的 id 等信息，该用户的所有行为，包括错题和收藏等，都将记录在对应的用户名下，实现小程序的个性化。

2.2 UML 相关需求分析图

首先不刷题小程序是一款面向所有小学生、小学教师、小学生的家长的刷题软件。用户注册登录进入软件，软件获取权限保存用户信息。小程序主要包含三

个主页面：主页，圈子以及我的。主页包含刷题，更换题集，记笔记，文字搜题，拍照搜索，打卡六个功能。圈子页面可以作为博主分享学习帖子，进入求助广场求助问题，关注问题或博主，收藏帖子。我的界面用于查看个人学习报告，浏览学习记录，查看收藏与错题集，更改个人信息。UML 缩略图如图 2.2.1 所示。

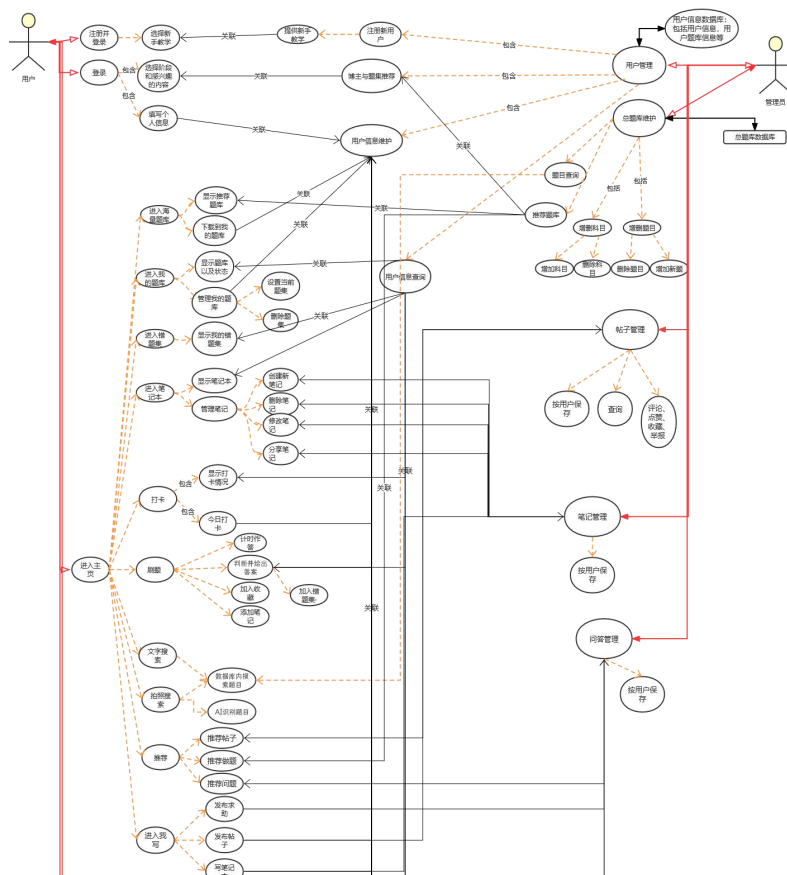


图 2.2.1 UML 缩略图

2.3 原型系统设计

在完成了功能设计和主要的界面构想后，我们使用墨刀设计了一个小型的原型系统，根据主要功能实现不同的用户交互界面，其中包括：用户登录界面、首页、我的界面、错题集界面、我的收藏界面和做题界面。

2.3.1 用户登录界面

用户登录界面位于我的界面里，当用户点击”点我登录”按钮后，会出现微信登录界面(如图 2.3.1)，用户可以通过微信进行登录。



图 2.3.1 微信登录界面

2.3.2 首页界面

我们首页的设计如图 2.3.2 所示，在首页的正上方有一些鼓励话语来鼓励正在刷题的学生，而在首页的下半部分则有三个按钮，分别是题库、正在刷题和我的收藏，用户可以通过点击不同的按钮进入到不同的界面。

(1) 用户点击“题库”进入题库界面，题库提供小学一到六年级语数外三个科目的题集供用户选择(如图 2.3.3 所示)，用户可在题库界面直接刷题，也可以将该题集添加到快捷入口“正在刷题”，之后可以直接进入“正在刷题”界面进行刷题。

(2) 用户点击正在刷题按钮时，用户会直接进入到做题的界面，而用户所做的题目则是用户在之前在做题时点击”设为当前题库”的题目。

(3) 用户点击我的收藏时会跳转到我的收藏界面。



图 2.3.2 首页界面



图 2.3.3 选择年级和科目的界面

2.3.3 我的界面

我的界面的设计如图 2.3.4 所示，黄色区域为用户登录退出区域，当前为未登录状态，点击登录按钮后实现登录。登录后使用微信小程序的 API 函数获取用户的 id、昵称、头像等信息，之后用户所有的行为都将记录在该 id 名下。在“我”界面中还可以选择自己的年级、进入我的错题集、我的收藏、学习报告界面。

(1) 用户点击年级时，会出现选择年级的选择栏(如图 2.3.5 所示)，用户选择完成后在年级那一行也会同时把用户的选择显示出来。

(2) 用户点击我的错题集和我的收藏时，会跳转到我的错题集和我的收藏界面。

(3) 用户点击学习报告时，会跳转到学习报告界面(如图 2.3.6 所示)，小程序将记录用户今日的和累计的刷题书以及学习时间，并在“学习报告”界面显示。



图 2.3.4 登陆成功界面



图 2.3.5 设置年级界面



图 2.3.6 学习报告界面

2.3.4 错题集界面

在错题集的界面中，用户可以选择不同的科目，当用户点击不同科目的时候，会跳转到错题界面(如图 2.3.8 所示)，在错题界面中会把用户在那个科目里的所有错题都显示出来。用户可以通过点击删除按钮把已经做会了的错题删除，同时也能通过点击上一题和下一题来查看不同的错题。

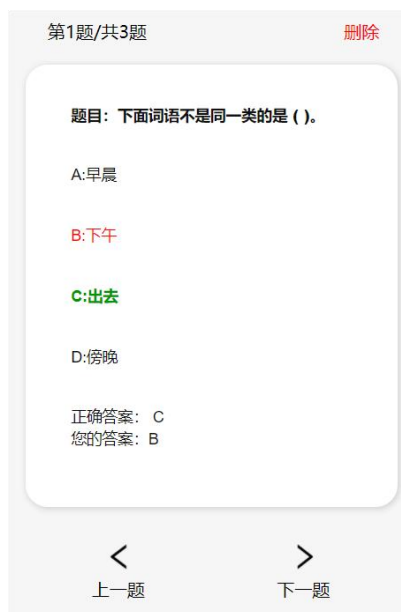


图 2.3.7 错题界面

2.3.5 我的收藏界面

我的收藏界面的设计如图 2.3.9 所示，在用户做题的过程中，若遇到认为有收藏价值的题目，可以点击收藏按钮，小程序将记录用户收藏的题目并在“我的收藏”界面显示，可以从主界面“我的收藏”进入，也可以从“我”界面中进入。如图 3.6 为收藏界面，显示用户收藏的题干，点击题干进入到详细题目页面，

如图 2.3.7 所示，正确答案用绿色显示，点击红色文字“删除”可删除该收藏的题目。

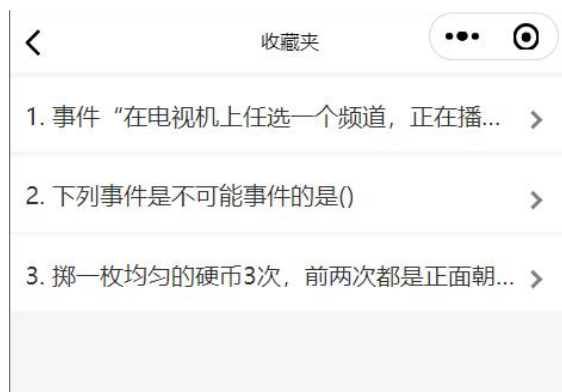


图 2.3.8 我的收藏界面

2.3.6 做题界面

做题界面的设计如图 2.3.10 所示，下方箭头可实现跳转到上\下一题，上方导航栏下有进度条，可以清晰地了解当前刷题进度。题目方框的左上方显示当前题号和总题数，右上方图标从左到右分别实现收藏、回到主页、添加到“正在刷题”，点击绿色按钮“提交答案”可提交当前答案，注意每道题目都要点提交答案。

当用户答完所有的题目后，会跳转到答题结果界面(如图 2.3.11 所示)，在完成界面中会显示用户的答题情况，用户可以点击不同的题目进入对应题目的详细界面(如图 2.3.12 所示)来查看正确答案



图 2.3.9 做题界面



图 2.3.10 答题结果界面

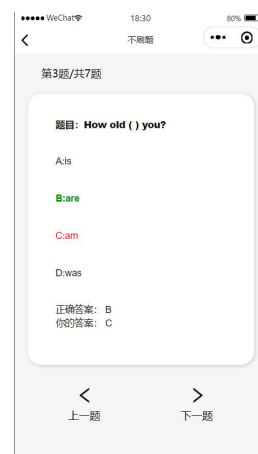


图 2.3.11 详细界面

3 概要设计和详细设计

3.1 系统结构

3.1.1 系统结构

整个小程序主要实现了用户登录与退出、刷题、收藏夹、错题集几个模块，几个模块相互联系，具体的联系具体可见数据库部分的详细阐述，此节主要介绍组成整个系统的几个模块，其功能模块的说明为：

1、用户登录与退出模块

用户可用微信账号登录小程序，小程序将记录用户所有的行为和状态上传至数据库，并在特定的界面显示，比如记录用户的错题，在错题集界面显示；记录用户收藏的题目，在我的收藏界面显示；自动生成用户的学习报告，在学习报告界面显示。用户退出不会删除数据库中的数据。

2、刷题模块

主页点击“题库”，由点击事件和页面跳转函数实现页面跳转，进入题集类型选择目录界面，提供了小学一到六年级语数外三个科目的题集供用户选择，点开一本题集即可显示刷题界面，选择一个答案点击提交，自动跳转到下一题，用户选择的答案会由 radio-group 组件的 change 事件传递给一个变量，通过比较该变量和正确答案以判断正误。做完题集后，显示所有题目的正误情况，点击对应的题干即可显示详细情况。刷题过程中可收藏题目、添加当前题集到“正在刷题”中。

3、收藏夹模块

在用户做题的过程中，若遇到认为有收藏价值的题目，可以点击收藏按钮，该题目被上传至数据库，点击进入“我的收藏”界面，使用云函数从数据库拉取对应用户 id 的收藏题目数组，依次显示。点击删除，通过云函数将该题从数据库中删除。

4、错题集模块

在用户刷题的过程中，若用户点击提交后小程序通过比较得知正确答案不等于用户选项，则将该题目上传至数据库，点击进入“错题集”界面，使用云函数从数据库拉取对应用户 id 的收藏题目数组，依次显示。点击删除，通过云函数将该题从数据库中删除。

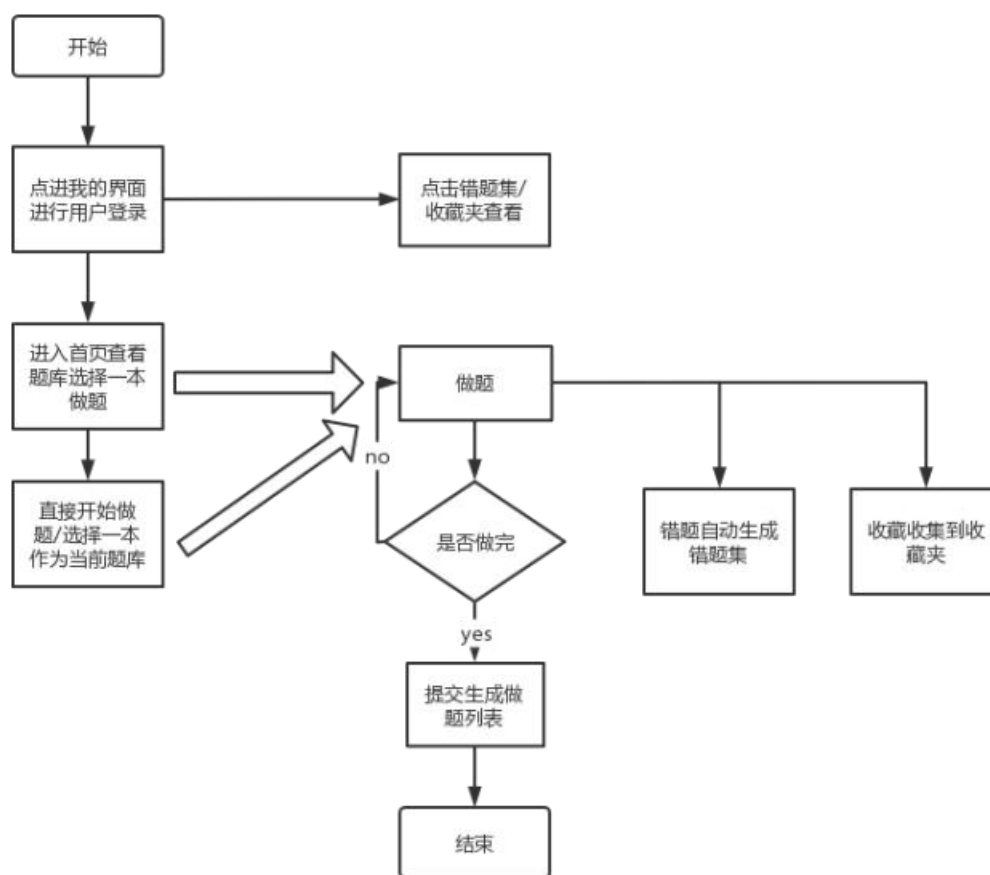


图 3.1.1 系统结构图

3.1.2 接口设计

在接口设计中，我们通过使用微信的云函数作为接口来实现界面和数据库的结合，其中我们用到的云函数包括：add 函数、get 函数、update 函数。我们在很多的地方都需要用到数据库中的数据，例如：

- (1) 当用户登录之后，我们需要把用户信息储存到数据库中，同时我们也需要把用户的id保存到全局变量中，以便之后修改数据库中的值，而要把用户的信息储存到数据库中，我们就需要使用add的云函数，在我们所设计的数据库中新增一行来保存用户的信息。
- (2) 当用户需要做题时，我们则需要在数据库把对应的题目取出来，所以用户点击不同年级和科目时，我们需要把用户所选的选项保存到全局变量中，然后通过使用get的云函数，把对应年级和科目的题目取出来。
- (3) 当用户做完题目后，我们需要把用户所做的错题保存到对应用户的数据

库中，所以这个时候我们就需要使用到update函数来更新数据库的内容。

3.2 类图等

运用类图等方法说明系统的设计。

从这一节开始自顶向下地讲解程序主体部分的设计与原理。主页对信息和功能进行综合，自顶向下包含题集，个人信息，收藏夹，题目，选项等结构，小程序主体 UML 类图如图 3.2.1 所示。

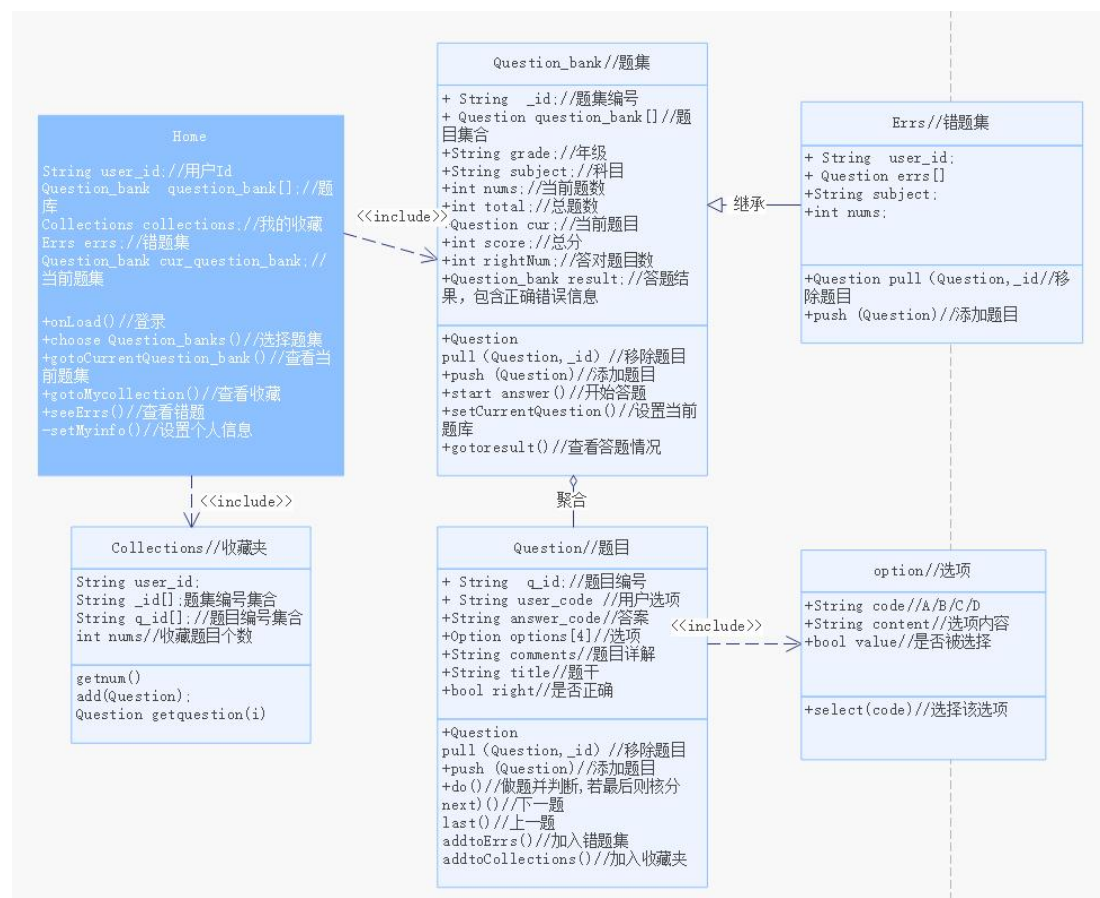


图 3.2.1 小程序主体 UML 类图

下面依次介绍这些类的功能及关系：

主页（Home）：

用户选择登录，处于未登录态时，用户可以选择题库答题，但软件并不会保存用户身份信息，学习记录等。登录时，需获取用户相关权限。登录后，用户可以进入题库选择题集进行答题，可以进入当前题库直接答题，可以查询学习报告，

查看收藏夹，查看错题集。

题集(Question_bank):

题集保存在云数据库，采用微信云开发 CMS 架构，依据不同学科不同年级分配题集编号。管理员有增删题库权限，会定期更新题库数据库，包括增加热点题目和删除冷门题，保持小程序高效性实用性。用户由对题集仅存在读取权限，防止恶意删改。

错题集(Errs):

错题集继承自题集，用户答题后会产生临时错题集，系统判断对错并核分使用户速览当前答题情况。永久性错题集保存用户每次在登陆状态下答错的题目，方便用户直接进入错题集开始答题，查漏补缺，用户也可随时将已经巩固完成的题目从错题集移除，保持错题集简洁高效。

收藏夹(Collections):

收藏夹用来保存用户答题时收藏的单个题目，用户有查询和删改权限，再主页可直接进入收藏夹答题。

题目(Question):

为方便管理与判断，题目一般为四个选项的选择题。每道题目有一个唯一的 `q_id` 题目编号(为在收藏夹和错题集中加以区分)，题干 `title`，四个选项 `options`，题目详解 `comments`。

选项(Option):

选项包含于题目，每个选项包含 `code`(A/B/C/D), `content` (选项具体内容)，`value`(是否被选择) 等信息。

3.3 关键数据结构定义

3.3.1 数据库数据

(1) 题库

题库保存了多个题集的具体的数据，用来为用户提供题目。在该小程序中，题集按照阶段和科目进行区分，因此每一条该数据集合中的每一条记录就是一本題集，需要保存的信息有：①`id`，用来独特的标志一个题集；②阶段和科目，方便检索；③题集中的题目，里面保存了一本題集中具体的题目；④数量，表示題集中题目的数量；⑤其他信息，在数据库创建一条记录的时

候自动生成，用来表示该条记录的一些操作信息。

该部分数据由小程序提供给用户，也就是说，需要开发者搜索数据并将其按照一定的格式存入数据库，数据变化流程图如图3.3.1所示。

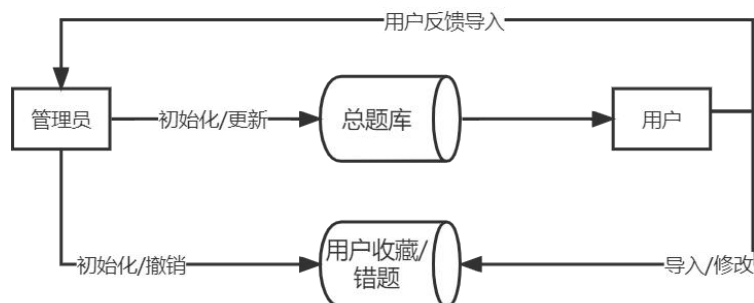


图3.3.1 数据库变化情况流程图

(2) 题目

题库中的一条记录为一个题库，而一本题集由数道题目组成，题目是批量加入到数据库的题库中去的，因此题目也有一定的结构。

一道题目包含的信息有：①id，用来独特地表示一本题集中的题目；②title，包含题目信息；③选项，包含选项的code比如A，和该选项的内容；④答案，该道题的正确答案；⑤解析，该道题目的答案解析；⑥用户选择，用来保存用户选择的答案，该项数据主要是为了与错题集统一，错题集需要保存用户的选项，在题库这个地方加入该项数据可以在push错题集的过程中只需要一步即可，方便了错题集的更新。

(3) 用户信息

用户需要拥有自己的相关设置，因此需要将用户信息保存在数据库中以便后续操作的时候进行区分。

用户信息包括：①id，用来独特的标志一个用户；②阶段，方便为用户推选题集；③头像和名字，用来更方便的检索用户；④相关信息，在创建一条记录的时候伴随着id一起创建的。

用户信息在用户进入小程序后点击“我的”之后进入用户操作信息的界面，点击登录后将自己的微信用户的信息选择传送给数据库，数据库将检索是否已经为该用户创建过记录，如果创建过，就不再创建，并将该用户对应的相关信息拉取到本地，否则如果是新用户，那么将为其在用户信息的集合

中新建一条记录保存他的信息，并在相关的数据库中为其创建需要的记录。

(4) 错题集

为了方便用户使用，错题集按照科目进行分类，并且错题集由用户单独拥有，因此需要将错题集按照用户id进行区分。因此对每个科目建立一个错题集合，其中的每一条记录为用户的该科目的错题集。

在该实现阶段，需要建立“Chinese”、“math”、“English”三个数据集。每个集合中的记录包含的信息为：①id，该条记录的标志；②open_id，用来与用户id进行匹配，以此找到对应用户的错题集；③错误集合，该用户在该科目中的错误题目合集，在用户提交答案并判定了之后直接自动加入；④数量，该题目合集的题目数量。

对于每条记录中的错误集合，它保存了所有的错题，是一个数组。其中数组的每一个元素包含：①id，对应题集中唯一标志一道题的id；②题目，该道题的题目；③选项，对应题目的多个选项和内容；④类型，该道题所位于的题库的名字，这里为阶段和科目；⑤答案，该道题目的正确答案；⑥用户选择，用户在这个位置所选择的选项；⑦解析，题目的具体解析。

错题集在用户第一次从创建的时候就会自动在三个错题集中新加一条记录，保证之后用户在做错题的时候可以直接将题目加入到该集合中，防止需要多次判断是否存在而浪费时间。

(5) 收藏夹

收藏夹保存了用户自己想要记录的题目的合集。

主要包含：①id，用来标志一条记录，为数据库自动生成；②openid，与用户的id相对应，保证可以检索到用户的收藏记录；③收藏列表，由三项数据组成，分别是题集id、题目id、具体题目，其中题集id和题目id是找到一道题目的关键码，用于检索题目，这样就不需要多次保存一个题目数据，从而节省了空间，而具体题目是为了方便展示整个收藏夹的题目，可以一次获取所有的题目然后整体展示到一个界面中去，可以先不检索，从而节省了时间，提升了用户体验。

收藏夹的记录在用户第一次从创建的时候就会自动创建，保证之后用户在点击收藏的时候可以直接将题目加入到该集合中，防止需要多次判断记录是否存在而浪费时间。

(6) 当前题库

当前题库为用户所选择的要刷的一本题集，因此只需要保存题集的标志性字段即可。

因此该结构所包含的信息有：①id，用来标志一条记录，为数据库自动生成；②openid，与用户的id相对应，保证可以检索到用户对应的记录；③阶段和科目，在本数据库中，题集的标志可以由阶段和科目确定，因此可以将此作为外码。

整个数据库中多个集合之间是有关联的，每个集合中的记录由数据库自动生成一个主码，同时也有具体含义的关键码，各个集合之间通过外码建立连接。

数据库总体模型如图3.3.2所示。

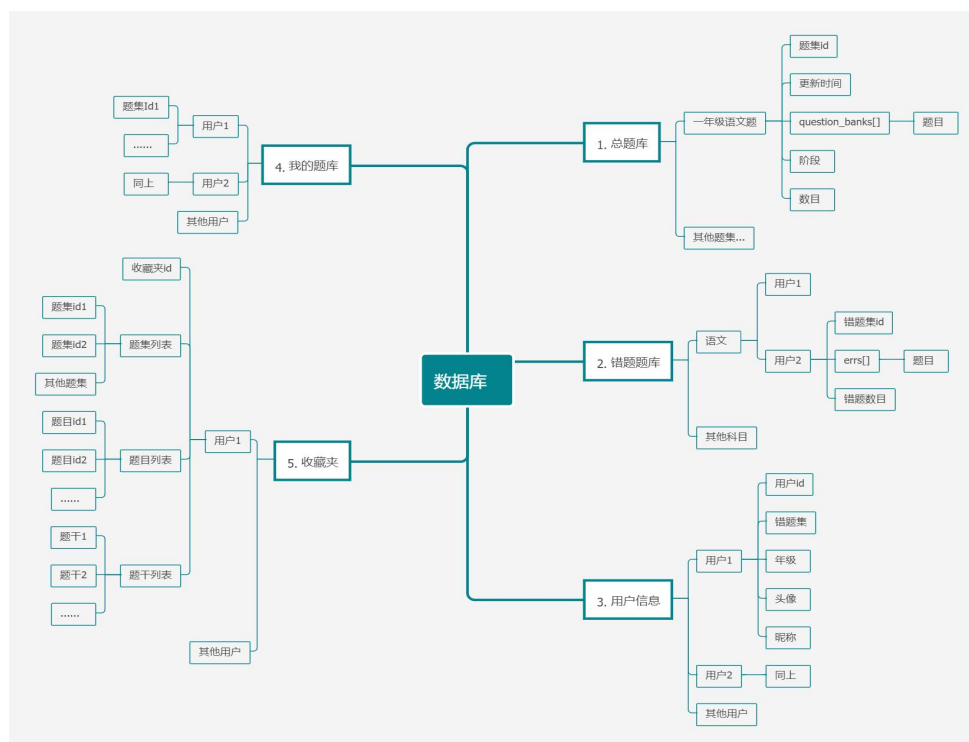


图3.3.2 数据库总体数据模型

3.3.2 后端数据

(1) 整个小程序的数据

小程序的全部变量数据可以帮助界面跳转时保存信息，并且可以保存一些小程序整体需要的信息。

保存的数据有：①用户id；②用户所处的阶段；③用户当前选择的科目；④用户点进去的题集；⑤用户的错题集；⑥用户的收藏集。这些信息在用户

点击界面跳转按钮是会保存相关的数据并将其传递给下一个界面，保证下一个界面可以收到用户想要的数据库数据并在数据库中检索出来。

(2) 单个界面数据

界面的全局变量和局部变量由该界面所实现的功能来决定，界面需要显示什么数据就在js文件中添加什么数据。

3.3.3 用户关联

用户在第一次进入小程序并登录后，在用户信息、错题集、收藏夹、当前题库中都会拥有自己的记录，并且会保存与用户相关的信息，其他的有关题库的部分会初始化为空值。

用户在登陆后，在自己做题的过程中对题目进行操作，小程序会将错题等数据直接保存到数据库中，并在本地保存一份，以此保证访问的速度。同时用户在点击相应的界面的时候会将需要拉取的数据库中的有关该用户的数据保存到本地，并保存下一个界面所需要的信息给整个小程序的全局变量中保证数据可访问。

3.4 关键算法设计

该小程序实现主要为前端渲染、后端、数据库几个部分。其中最重要的是后端部分，即响应函数的设置，这些函数在关键函数说明中已经进行具体的解释，在这里主要描述一下小程序在整体设计和实现过程的流程。

整个小程序的设计流程如图所示，app中的整体配置可以将小程序共有的一些属性和数据进行初始化，从而减少了具体界面设置的次数，同时还需要设置首页，也就是用户第一次打开小程序所看到的界面。

另外一个重要的点在于界面的跳转，界面设置了之后需要将其链接到需要跳转的地方，才能保证该界面被点进去。在app整体配置中可以配置导航栏的跳转，但具体界面的跳转需要在具体界面中配置，这个在界面设计章节中有详细的解释。

整体配置设置完毕后，需要进入子界面的设计。

在子界面设计中，需要首先获取到需要的数据库数据和app全局变量的数据到本地界面处，保证访问时的速度。之后需要根据界面所要实现的具体功能对其进行相应函数的编写，这一部分在关键函数设计这一章节有具体的解释，这里不再赘述。

在完成子界面的设计后，需要根据数据的含义将需要修改的全局数据进行修改，保证在界面跳转之后得到的是正确的数据。

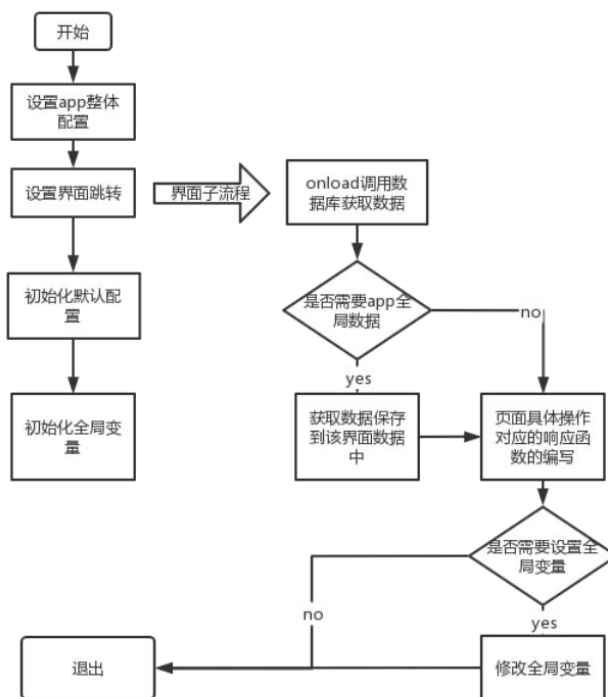


图3.4.1 整体设计流程图

3.5 数据管理说明

小程序所涉及到的所有数据都会在数据库中进行保存，包括用户的个人信息、所有题库、用户所拥有的单独的错题集、收藏集、当前题库等等，也就是说整个小程序都是基于该数据库实现的，而程序只是将这些数据以更好更方便的形式呈现给用户。

数据库的数据操作包括基本的增删改查操作、数据库的整体导入、数据结构的设计和 content 管理。

3.5.1 增删改查

整个数据库操作是基于微信开发者所提供的数据库实现的，微信开发者也提供了数据库的接口，可以采用其提供的云函数中关于数据库的函数对数据库进行操作。

1. 增加集合

增加一个集合的操作可以从数据库中通过可视化界面直接进行添加，在该数

数据库对应的云开发中的内容管理平台可以通过构造数据结构来定义一个数据库中的集合的数据结构，构造成功后，在云开发管理平台可以直接使用该结构添加一个集合。之后在后端中就可以直接使用该集合。

2. 增加记录

向集合中添加一条记录，可以使用数据库提供的add函数，即 `db.collection().add()`，便可以向集合中添加一条记录。

3. 删除数据

删除集合中数组的一条数据，可以使用update函数，在该小程序中，删除的数据并非为一条记录，因为一条记录保存着一个用户的所有题目的信息，所以删除的为一条记录中的数据的一个数据，因此需要采用update函数对该条记录进行更新，首先获取到该项数据，在本地将该删除的数据进行删除，然后update到数据库中。

4. 更改记录

更改数据库中的一条记录，同样采用update函数，首先将需要更改的记录查询出来保存在本地的一个数据项中，在本地对其进行所需要的具体的修改操作，修改后将其通过update函数赋值给数据库中对应的数据项，就可以得到新的数据。

5. 查询记录

查询一条数据需要采用get函数，由于在用户获取数据的时候往往是获取一条记录，因此采用 `db.collection().doc().get()` 函数将想查询的数据保存到本地。

3.5.2 批量数据导入

数据库导入新题集流程：

- (1) 从开放题库中找到图片(或直接为 word 类型)题目，识别转换为 word。
- (2) 得到 word 文本题集后，按照一定规则规范格式，将题干与选项之间，选项与选项之间，选项与答案之间，答案与解析之间插入换行符，最终格式如图 3.5.1 所示。

1. 事件“在电视机上任选一个频道，正在播放天气预报节目”是0
 - A. 必然事件
 - B. 不确定事件
 - C. 确定事件
 - D. 不可能事件
2. 下列事件是不可能事件的是0
 - A. 数轴上的数右边的总比左边的大
 - B. 随便翻开数学七年级（上）课本，一下翻到 88 页
 - C. 我国沿海每年都会刮台风
 - D. 小周口袋里有两个黄乒乓球，可他任意一摸，却摸出一个白乒乓球

图3.5.1 最终格式

（3）转换格式之后，全选文字，点击‘插入’‘将 word 转成表格’,设置列数和文字分隔位置，具体设置如图 3.5.2 所示。

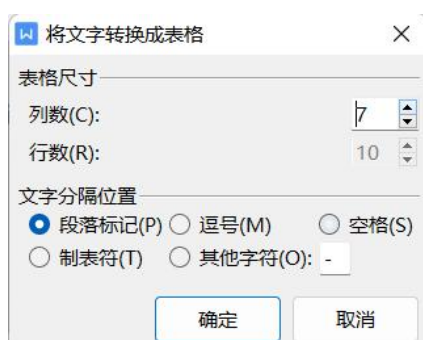


图3.5.2 具体设置

（4）转换完成后，全文复制到 excel,保存，进入 [JSON 生成工具-v7 \(xiaomutong.com.cn\)](https://xiaomutong.com.cn)链接，导入 excel，即可生成 json 文件。

（5）最后在微信云开发平台导入 json 格式的题集，如图 3.5.3 所示。



图3.5.3 导入json格式的题集

3.5.3 内容管理

数据库中的数据一方面由我们开发者在内容管理平台进行结构定义和批量导入，另一方面可以通过云开发管理平台进行记录的直接的可视化的增删改查的操作。

另一方面小程序的后端会调用数据库并对其进行相关的操作来完整的呈现一个功能，所以相关数据库函数调用产生的效果要与在内容管理平台所定义的操作的结果保持一致，因此在写后端程序的时候，要与数据库管理人员相结合商讨最佳的数据结构，不仅保证数据存储，而且要方便开发者调用，同时我们开发者还需要保证所增删改查的格式与定义相同，保证数据的一致性。最后我们设定了前文所述的数据结构，并采用了上述的数据库操作函数来完成相关功能。

3.6 界面设计

3.6.1 概要设计

(1) 界面生成：在miniprogram中的pages文件中保存了所有的界面文件，可在app.json中添加新的界面，保存后在pages文件中自动生成该界面。

(2) 界面的构成与实现：每个界面文件由js文件、json文件、wxml文件、wxss文件构成。js文件用于前端设计，包括界面所需变量的设置、界面的生命周期函数、点击事件的处理函数、与数据库交互的云函数等等。wxml和wxss文件类似网页的html和css，用于实现界面的渲染、布局 and 美化。

(3) 界面的跳转：以刷题界面为例，用户在题库界面选择一个题集，用bindtap功能为该题集设置一个点击事件，在题库界面的js文件中编写点击事件对应的函数，使用wx.navigateTo函数跳转到目标界面，这样用户点击这个题集便能进入到对应的刷题页面。如果要跳转到tabbar中的界面，则用wx.switchTab函数

(4) 界面的静态显示：在wxml文件中用view、text等组件静态显示特定的内容，在wxss中进行美化

(5) 界面的动态渲染：以刷题界面为例，进入该页面后要显示某阶段某科目的第一道题目，则该题目是个变量，在wxml中用{{}}包含该变量，并在js文件中用云函数读取数据库中的第一道题赋值给该变量，即可在刷题界面动态渲染该题目。点击下一道题，js中对应的点击事件函数将题目变量设为下一道题，即可实现题目的转换。点击提交，js中对应的点击事件函数会进行相应的处理，记录用户的正确情况，并跳转到下一题

3.6.2 详细设计

(1) 运用微信小程序开发平台提供的组件和API

微信小程序开发平台提供了丰富的组件和API，可以实现许多功能。每个组件都有许多属性可以自行设置，以产生不同的效果，比如view组件有hover-class属性，可以实现点击态效果。我们的小程序中主要用到了view、text、image、swiper、radio组件。text组件可以显示文本内容；view组件可用作视图容器，主要功能是容纳其他组件以实现组件的布局，从而将页面分割为独立的、不同的部分，它也可以作为单独的组件显示文本内容，如图3.6.1为主页的三个选项的代码实现，文本内容由text组件实现，多个text组件被view组件层层包裹，形成多个容器，每个容器都可以设置不同的样式，从而形成能如图3.6.2的页面布局；image组件用于显示图片，可以在它的src属性中指定图片的路径，可以是本地图片路径也可以是网络图片的地址，mode属性可以设置图片裁剪、缩放的模式；swiper组件用于和image组件一起实现轮播图，可以在属性中设置是否显示面板指示点、指示点颜色、当前选中的指示点颜色、是否自动切换、自动切换时间间隔等效果，如图3.6.3为轮播图实现代码，imageList是一个图片路径字符串数组变量，用for循环将数组元素依次渲染到轮播图中，for循环渲染之后会详细讲解；radio组件用于实现单选选择器，是答题界面的核心组件，一个radio组件控制一个选项的选择，通过设置每个选项的value属性，可以在用户点击选择时获取所选选项的value值，用于判断正误。

```
<view class="content1">
  <view class="tiku" bindtap="gotiku">
    <view>
      <text class="tikutext">题库</text>
      <text class="right_arrow"></text>
      <text class="engl"> Collections</text>
    </view>
  </view>
  <view class="mytiku" bindtap="gomytiku">
    <view>
      <text class="mytikutext">正在刷题</text>
      <text class="right_arrow"></text>
      <text class="engl"> Current</text>
    </view>
  </view>
  <view class="likes" bindtap="goToCollection">
    <view>
      <text class="myliketext">我的收藏</text>
      <text class="right_arrow"></text>
      <text class="engl"> Likes</text>
    </view>
  </view>
</view>
```

图 3.6.1 主页的三个选项的代码实现



图 3.6.2 首页页面布局

```
<swiper indicator-dots autoplay interval="2000" circular class="swiper">
  <swiper-item wx:for="{{imageList}}" class="what">
    <image src="{{item}}" class="pic"></image>
  </swiper-item>
</swiper>
```

图3.6.3轮播图实现代码

(2) 界面管理和及其整体风格的实现、底部导航栏tabbar的实现

打开全局文件app.json，可以在“pages”:[]中添加和删除界面，如图3.6.4为所有界面；在“window”:{}中设置所有界面的颜色、导航栏的标题等等，如图3.6.5，如果想改变特定界面的风格，可在对应界面的json文件中修改；“tabBar”:{}中设置底部导航栏，可设置背景颜色、标题、图标、选中和未选中的颜色、跳转界面路径等等，如图3.6.6

```
"pages": [
  "pages/home/home",
  "pages/danxuan/danxuan",
  "pages/errorclt/errorclt",
  "pages/errorlist/errorlist",
  "pages/me/me",
  "pages/tiku/tiku",
  "pages/tikuxiaoxue/tikuxiaoxue",
  "pages/tikuxiaoxue1/tikuxiaoxue1",
  "pages/xiaoxue1math/xiaoxue1math",
  "pages/mytiku/mytiku",
  "pages/collection/collection",
  "pages/questiondetail/questiondetail",
  "pages/record/record",
  "pages/currentErr/currentErr",
  "pages/currentErrdetail/currentErrdetail"
],
```

图3.6.4 所有界面

```
"window": {
  "backgroundColor": "#F6F6F6",
  "backgroundTextStyle": "light",
  "navigationBarBackgroundColor": "#FFFFFF",
  "navigationBarTitleText": "一起来答题",
  "navigationBarTextStyle": "black"
},
```

图 3.6.5 window 的 css

```
"tabBar": {
  "color": "#cdcdcd",
  "selectedColor": "#333333",
  "backgroundColor": "#ffffff",
  "list": [{
    "pagePath": "pages/home/home",
    "text": "刷题",
    "iconPath": "images/home-no.png",
    "selectedIconPath": "images/home-yes.png"
  }, {
    "pagePath": "pages/errorlist/errorlist",
    "text": "错题集",
    "iconPath": "images/error-no.png",
    "selectedIconPath": "images/error-yes.png"
  }, {
    "pagePath": "pages/me/me",
    "text": "我的",
    "iconPath": "images/me-no.png",
    "selectedIconPath": "images/me-yes.png"
  }
]
```

图3.6.6 tabBar代码实现

(3) 界面动态渲染

界面的动态渲染即在界面上显示一个变量，显示出来的值跟随变量的值变化，而不是一个固定的值。如图3.6.7是答题界面上方的进度信息显示代码，由于所有阶段和科目的答题界面是同一个界面，所以当前的题号current和总题数total显然是变量，这些变量在js文件中的data:{}中定义，total变量通过读取对应阶段科目的数据库获得，current变量通过点击提交按钮的次数获得，在wxml中用两对大括号包裹在js中定义的变量，即可实现动态渲染。


```
<view class="content">
<view class="num">第{{current}}题/共{{total}}题</view>
```

图3.6.7进度信息显示代码

(4) wx:for 循环界面渲染实现每道题目的显示

如图3.6.8为显示题目的代码实现，subject为js文件中的对象变量，代表每一道题目，它的数据成员title、options分别表示题干、选项数组，选项数组的每个元素是一个对象，它的数据成员code、content分别表示选项码（A、B、C、D）和选项内容。

radio组件是一个单选项组件，其value属性是radio 标识，当该radio选中时，radio-group的change事件会携带radio的value，checked属性表示当前是否选中，根据它可以实现选中 and 未选中的显示效果，如图3.6.9。radio-group是单项选择器，内部由多个radio组成，其bindchange属性可以实现当radio-group中选中项发生改变时触发change事件，detail={value:[选中的radio的value的数组]}，这样就可以在js文件中利用detail.value（用户选择的选项）和正确选项对比来判断正误。isSelect是js中的变量，初始化为false，没做完一道题点击提交后也将其设为false，这样可以使题目显示出来后所有选项都是未选中的状态。

首先用view组件显示题干，用radio-group组件包裹四个选项，在radio-group组件中，通过wx:for="{{subject.options}}"实现循环遍历选项数组，item代表当前循环的数组元素，每次循环中，实现一条选项（选项码+选项内容）的显示，以及对应radio选择器的属性的赋值。

```
<view class="question" >题目: {{subject.title}}</view>
<radio-group bindchange="radioChange">
  <view wx:for="{{subject.options}}" class="option">
    <radio value="{{item.code}}" checked="{{isSelect==item.code}}"/>
    <text class="option2">{{item.code}}:{{item.content}}</text>
  </view>
</radio-group>
```

图3.6.8显示题目的代码实现



图 3.6.9 选中和未选中的的显示效果

(5) wx:if 实现条件显示

userInfo保存用户信息，包括id、微信昵称、头像等，初始化为空，用户点击登录后通过微信提供的API实现用户id的获取，保存在userInfo变量中。通过wx:if="{{!userInfo}}"判断当前是否有用户登录，如果没有，显示点我登录按钮，否则（wx:else）显示用户头像、微信昵称和退出登录按钮，如图3.6.10。

```
<view wx:if="{{!userInfo}}" class="root">
  <view class="circle"></view>
  <button class="login2" bindtap="login">点我登录</button>
</view>
<view wx:else class="root">
  <image class="icon" src="{{userInfo.avatarUrl}}"></image>
  <text class="nickname">{{userInfo.nickName}}</text>
  <button bindtap="logout">退出登录</button>
</view>
```

图3.6.10 wx:if实现条件显示

(6) 页面跳转的实现

在这个小程序中，我们主要用了三种的跳转方式，分别是wx.navigateTo、wx.switchTab和wx.reLaunch。

(a) wx.navigateTo的主要功能是保留当前页面，并跳转到应用内的某个页面。在我们的程序中，大部分的界面跳转都是使用wx.navigateTo，用户可以通过点击左上角的返回标志(如图3.6.11所示)返回上一页。

(b) wx.switchTab的主要功能是跳转到 tabBar 页面，并关闭其他所有非 tabBar 页面。在我们的程序中，我们的tabBar中的页面分别有首页，错题集页面和我的页面，所以当我们跳转到这些页面时，我们就要使用wx.switchTab函数。

(c) wx.reLaunch的主要功能是关闭所有页面，打开到应用内的某个页面。当我们不想一直点击返回才能返回首页时，我们就可以使用这个函数，这样用户可以通过点击左上角的返回首页标志(如图3.6.12所示)直接返回首页。



图3.6.11 返回标志



图3.6.12 返回首页标志

(7) 点击事件的实现

在我们的小程序中，我们通过bindtap来实现点击事件，bindtap的功能是当用户点击某一个组件时会在该页面对应的 Page 中找到相应的事件处理函数并执行，所以我只需要在对应页面的Page里的事件处理函数中写下我们需要实现的功能对应的代码，就可以实现点击事件。下面我们答题结果页面为例。

在答题结果页面中，我们通过for循环把用户刚做完的所有题目都显示出来，用户可以通过点击不同的题目跳转到对应题目的详细解答中，而要实现这样的点击事件，我们就需要知道用户点击的是哪一道题，所以我们不单单只需要事件处理函数，我们还需给它传参，我们通过data-index来把用户点击的题号传进事件处理函数中(如图3.6.13所示)，然后在事件处理函数中把题号保存到全局函数中(如图3.6.14所示)，这样就成功的完成了一件点击事件。

```
<view bindtap="seeQuestion" data-index="{{index}}" class="title">{{index+1}}: {{item.title}}</view>
```

图3.6.13参进事件处理函数

```
seeQuestion(e){
  console.log('e.currentTarget.dataset.index',e.currentTarget.dataset.index);
  app.globalData.globalCollectionNum=e.currentTarget.dataset.index
  wx.navigateTo({
    url: '/pages/currentErrdetail/currentErrdetail',
  })
}
```

图 3.6.14 事件处理函数

(8) 在wxss文件设置界面样式，实现界面美化

在组件名后添加class="", 设置样式名，在wxss文件中用. 样式名 {} 设置想要的样式。常见的样式设置有组件水平布局、垂直布局、文字居中显示、盒子模型（如图3.6.15所示）

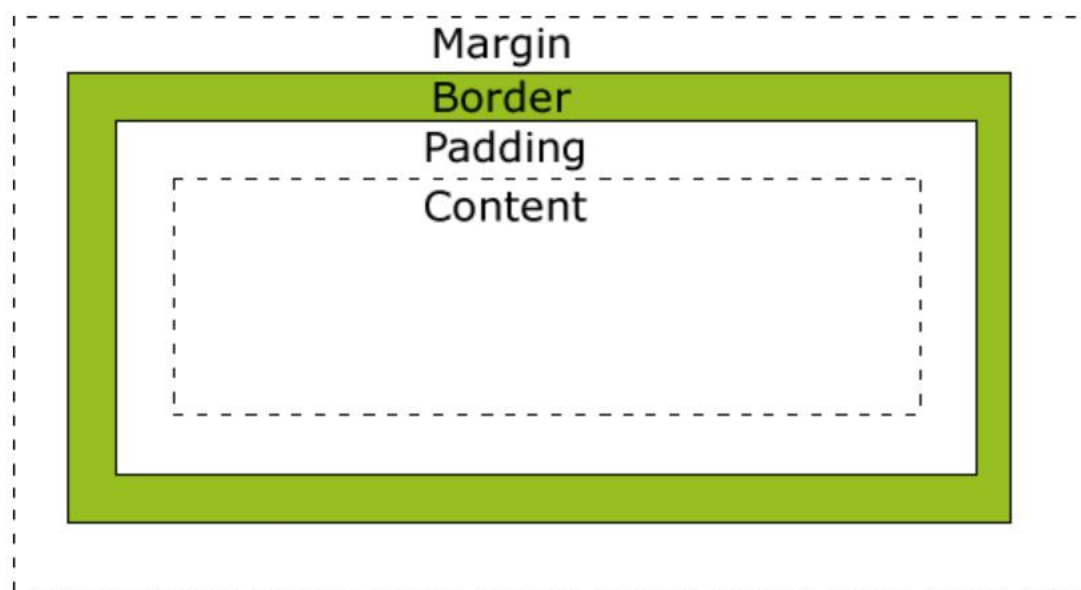


图3.6.15 盒子模型

4 实现与测试

4.1 实现环境与代码管理

4.1.1 实验环境

Windows10/11+微信小程序开发工具进行开发，数据库采用微信开发者提供的云开发工具，主要为数据库，在内容管理平台和云开发管理平台两个平台对云数据进行管理。

4.1.2 代码管理

1. 微信开发者协同代码工作平台

第一个是采用微信开发者提供的协管代码工作平台进行管理。微信小程序开发工具提供了一个协同代码开发功能，在该工具中的版本管理中可以推送自己的代码，也可以拉取团队成员推送的代码。

共同负责前端的两位同学可以将自己完成的界面推送到微信所提供的代码管理平台并合并，另一个同学可以拉取下来，并直接将代码合并到自己目前的项目中，实现两人工作的合并，非常方便，代码管理界面如图4. 1. 1所示，管理成员如图4. 1. 2所示。

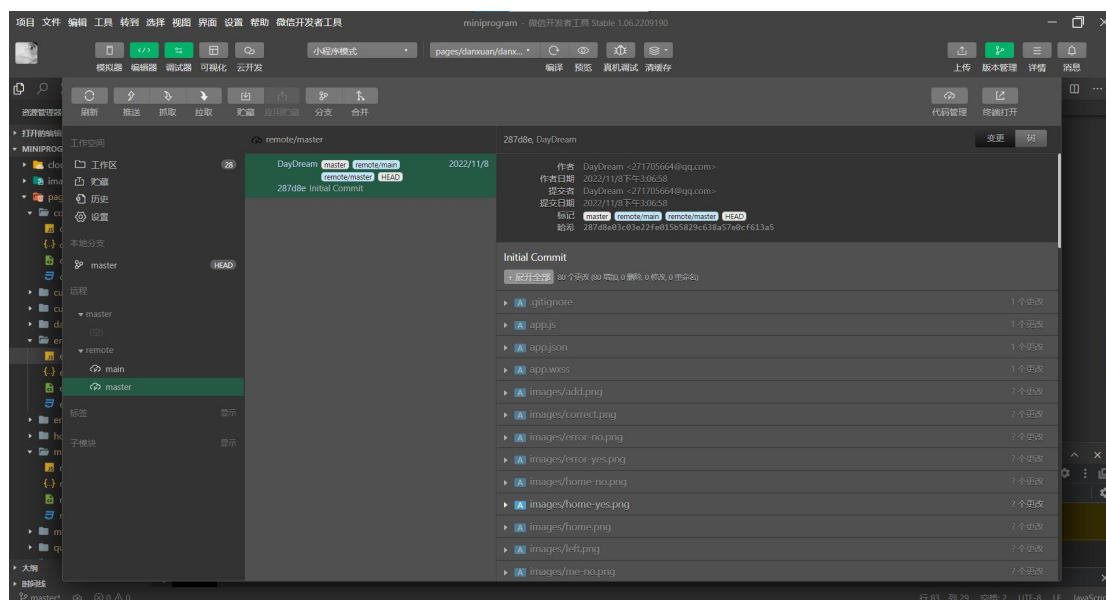


图4. 1. 1 代码管理界面



图4. 1. 2 管理成员列表

2. gitee协同开发管理

另一个是采用gitee协同开发管理。可以用于代码托管和团队协作。我们采用该平台保存了代码并可以向外展示。

在本次的协同开发，主要采用了微信开发者提供的代码托管平台，gitee作为辅助，git代码托管平台文件列表如图4. 1. 3所示。

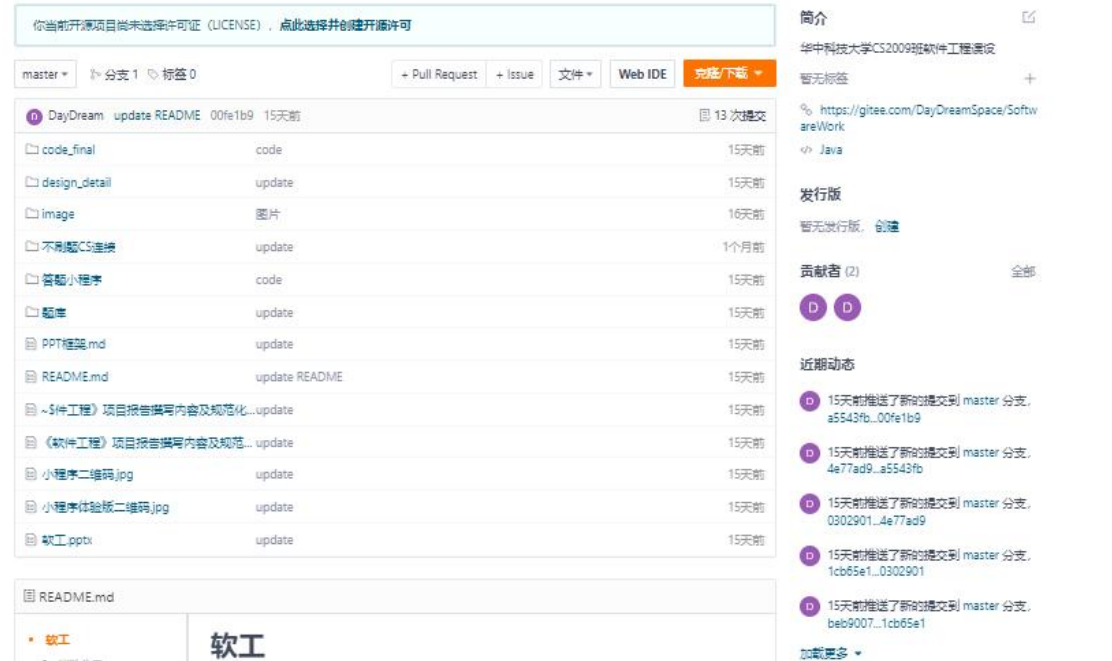


图4. 1. 3 git平台代码列表



图4.1.4 gitee管理图

4.1.3 小程序发布管理

小程序最后发布上线，并且可以不断修复bug重新发布，因此可以在微信提供的小程序注册平台对小程序进行注册并将本地的小程序项目与其关联，关联后便可以将代码上传到该平台，上传完成后，在小程序发布管理平台对其进行审核提交和发布，最后小程序便可以上线，并且可以用于其唯一的二维码方便用户扫描进入，版本管理过程如图4.1.4所示。

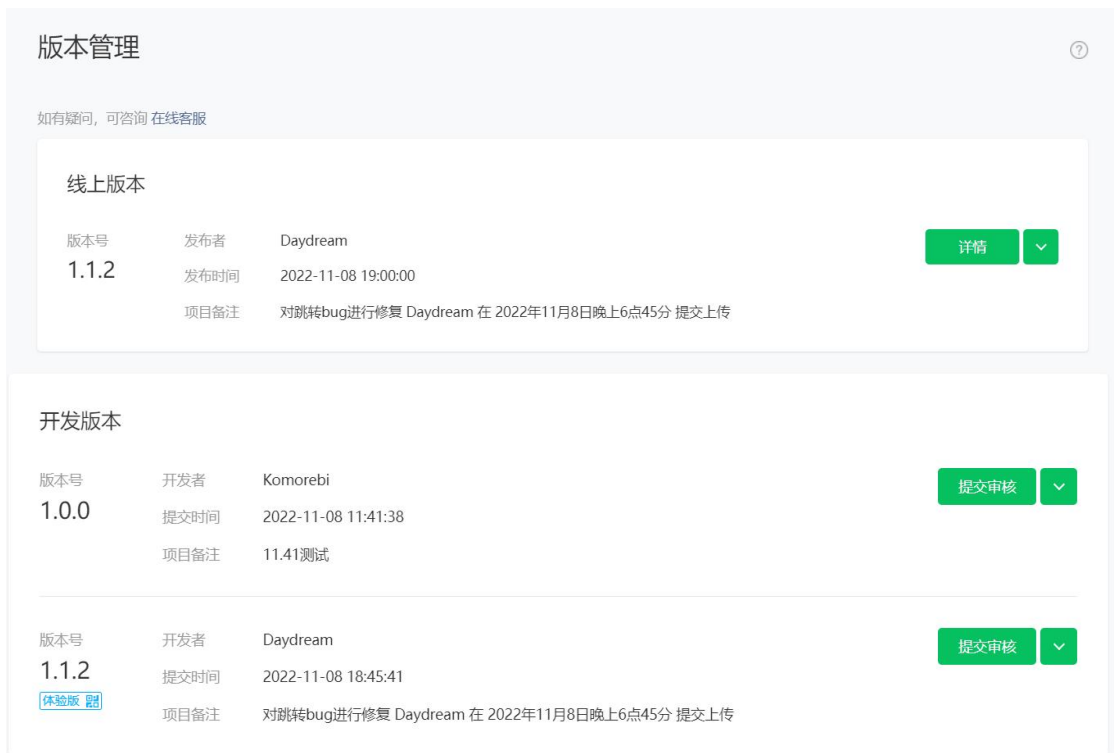


图4.1.5 版本管理过程

4.2 关键函数说明

1. 单个页面关键函数

登录界面:

`login()`: 用户注册并登录函数。利用云函数 `wx.cloud.callFunction()` 获取授权, 获取用户微信 `id`, 在数据库的用户表中添加该用户, 并且创建属于该用户的存储区, 用户答题信息, 身份信息在每次登录时都保存在云数据库中。

`logout()`: 退出登录

答题界面:

`onload()`: 答题界面初始化函数。需要从数据库中读出所选题库信息, 并在屏幕显示第一道题的答题界面, 等待用户答题。

`radioChange(e)`: 用户点击响应函数。一旦用户选择选项, `isSelect` 置 1, 接收用户答案保存在 `userSelect`, 用于后续检验正误。

`submit()`: 提交函数。如果用户在未答题情况下提交, 创建窗口提醒用户先进行答题, 并进行响应异常处理。如若正常提交, 匹配用户选择选项与答案选项, 判断正误, 如若正确, 累计计分, 如若错误, 将该题目加入错题集, 并保存答题结果。另外, 如果当前题目是最后一道题, 直接跳转到答题结果界面, 显示答题情况。

`next()/pre()`: 切换题目函数。答题时, 切换上一题或下一题, 并在第一题和最后一题时做出提示。

`addmytiku()`: 加入到我的题集。将当前题集设置为当前题集, 将信息保存在数据库。

`collect()`: 收藏。收藏当前题目到我的收藏夹, 依据用户 `id` 导入数据库该用户的收藏夹。

错题集界面:

`onload()`: 错题集初始化函数。依据用户 `_id` 和所选科目, 从数据库读取用户当前科目错题, 在页面显示用户选择的答案和错题解析。

`next()/pre()`: 切换题目函数: 切换上一题或下一题, 并在第一题和最后一题时做出提示。

收藏界面:

`onload()`:收藏夹初始化函数。依据用户 id 从数据库读取当前用户收藏题目, 加题干信息排列显示在主界面。

`seeQuestion()`:查看题目详情函数。依据用户点击的题干进入题目详解。

2. 界面交互关键函数

`question_banks()`:进入题库选择界面, 依据用户选择设置题目阶段科目

`gotoCollection()`:跳转到我的收藏界面, 依据用户身份_id, 从数据库读取当前用户收藏题目, 并显示。

`gotoMyQuestionbank()`:直接进入我的题集答题界面

`gohome()`: 从各个界面直接跳转到主页。

3. 数据库接口关键函数

本项目基于微信云平台开发, 云开发数据库存储数据, 使用了大量有关云接口函数, 例举关键函数如下表 4.2.1。

表 4.2.1 云函数表

云数据库初始化	<code>cloud.init({env: 'test-x1dzi'})</code>
获取用户微信名及 id	<code>wx.cloud.callFunction({name: 'getopenid'})</code>
导出数据记录	<code>wx.cloud.database().collection().get().then(res=>{})</code>
更新数据记录	<code>wx.cloud.database().collection().doc().update({ data:{}})</code>
在数据库创建新记录	<code>wx.cloud.database().collection().add()</code>
向数组中添加数据	<code>wx.cloud.database().collection().update({ data:{db.command.push()}})</code>

4.3 测试计划和测试用例

1、本地测试

我们通过本地的函数的输出来检查数据是否成功的存进数据库或者成功的把数据从数据库中读出。

(1) 我的界面

(a) 检查是否成功获得用户的基本信息(如图4.3.1), 如: 用户id、昵称等。



图 4.3.1 检查是否成功获得用户的基本信息

(b) 检查是否成功创建用户的错题集等信息(如图4.3.2所示)

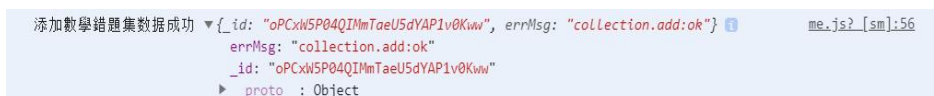


图 4.3.2 检查是否成功创建用户的错题集等信息

(2) 做题界面

检查用户选项，如果用户做错是否写进错题集(如图4.3.3所示)



图 4.3.3 检查用户选项

(3) 答题结果界面

检查用户所要查看的题目的题号(如图4.3.4所示)



图 4.3.4 检查用户所要查看的题目的题号

(4) 我的错题集界面

检查是否成功删除错题集数据库中的对应题目(如图4.3.5所示)



图 4.3.5 检查是否成功删除错题集题目

(5) 我的收藏界面

检查是否成功删除收藏数据库中的对应题目(如图4.3.6所示)



图 4.3.6 检查是否成功删除收藏题目

2. 整体的调试

通过在微信小程序的仿真器上，可以查看到实时的效果。在写前端代码时，我们可以通过观察仿真器上的效果来对我们代码进行修改，从而得到一个美观的前端界面，初始的做题界面和修改后的做题界面如图4.3.7所示。

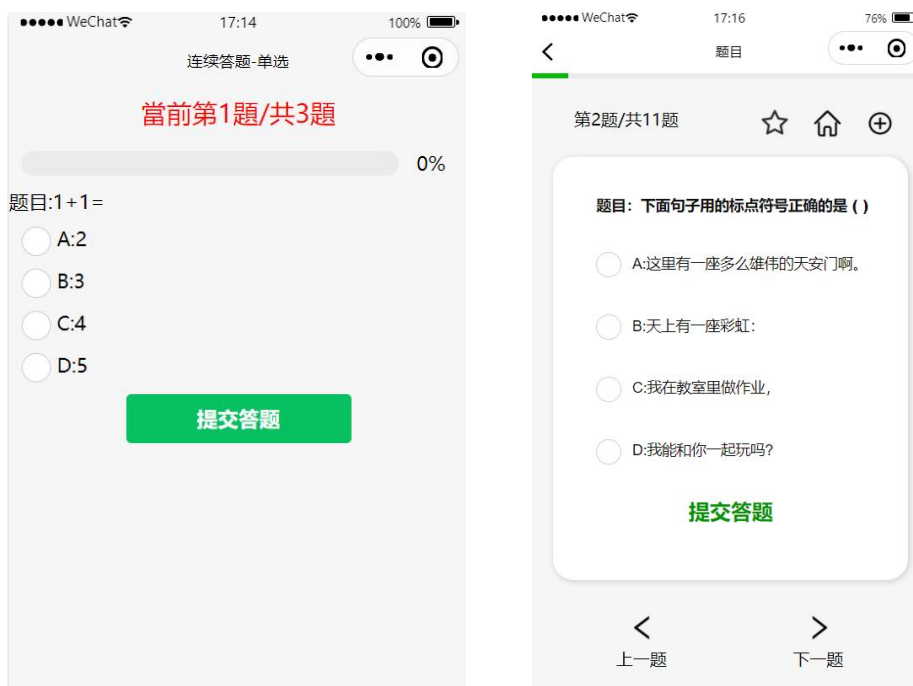


图 4.3.7 初始的做题界面(左边)和修改后的做题界面(右边)

3. 管理员体验版调试

通过在手机上使用体验版，我们发现在不同的机型下，前端的界面会发生一些的变化，在较大的机型下，前端的界面会出现右边有白边的问题，在较小的机型下，前端界面会出现显示不全的问题。经过我们在不同机型下的调试，我们知道了导致这个问题的原因是因为我们在写wxss文档时，我们把所有的距离的单位

都设为了px而不是rpx，所以界面不能随着机型大小的变化而变化。

4.4 结果分析

用户登录与退出：能正常实现，用户登录后可以获取该用户的 id、微信昵称、头像等信息并能上传至数据库，退出后信息保留。

题集的选择与刷题界面的显示：进入“刷题”界面可以正常选择阶段和科目，刷题界面能够正常显示题目和选项，点击上\下一题可以正常跳转，点击提交按钮答案可以正常保存，做完题集后正确情况可以正常显示，点击可以正确显示对应的题目和用户选项。

错题记录与删除：初始状态下或未登录状态下正确显示“没有错题”，点击“去练习”按钮正确跳转到“正在刷题”界面。刷题过程中能够正确记录用户的错题和错误选项到数据库，并在错题集界面的不同科目分类下正确显示。点击删除后错题可以正常从数据库中删除。

收藏与删除：初始状态下或未登录状态下正确显示“没有收藏的题目”，点击“去练习”按钮正确跳转到“正在刷题”界面。刷题过程中点击收藏可以正常保存到数据库。并在收藏界面中正确显示。点击删除可以正常从数据库删除。

添加到“正在刷题”：初始状态下或未登录状态下正确显示“没有添加正在刷的题集”，点击“去添加”按钮正确跳转到“刷题”界面。刷题界面点击添加图标能将题集正确添加到“正在刷题”界面。

5 总结

5.1 用户反馈

5.1.1 优点

该小程序为用户提供了随时随地做题的功能。在上线之后广受好评。并且在微信小程序的搜索框中可以直接被搜索到，很方便用户查找并使用。如图 5.1.1 所示，我们的小程序在搜索结果中排为第一位。



图 5.1.1 “不刷题”小程序搜索结果

在上线之后，有许多用户搜索小程序并使用，如图 5.1.2 所示即为上线后注册小程序的人数和活跃用户数。

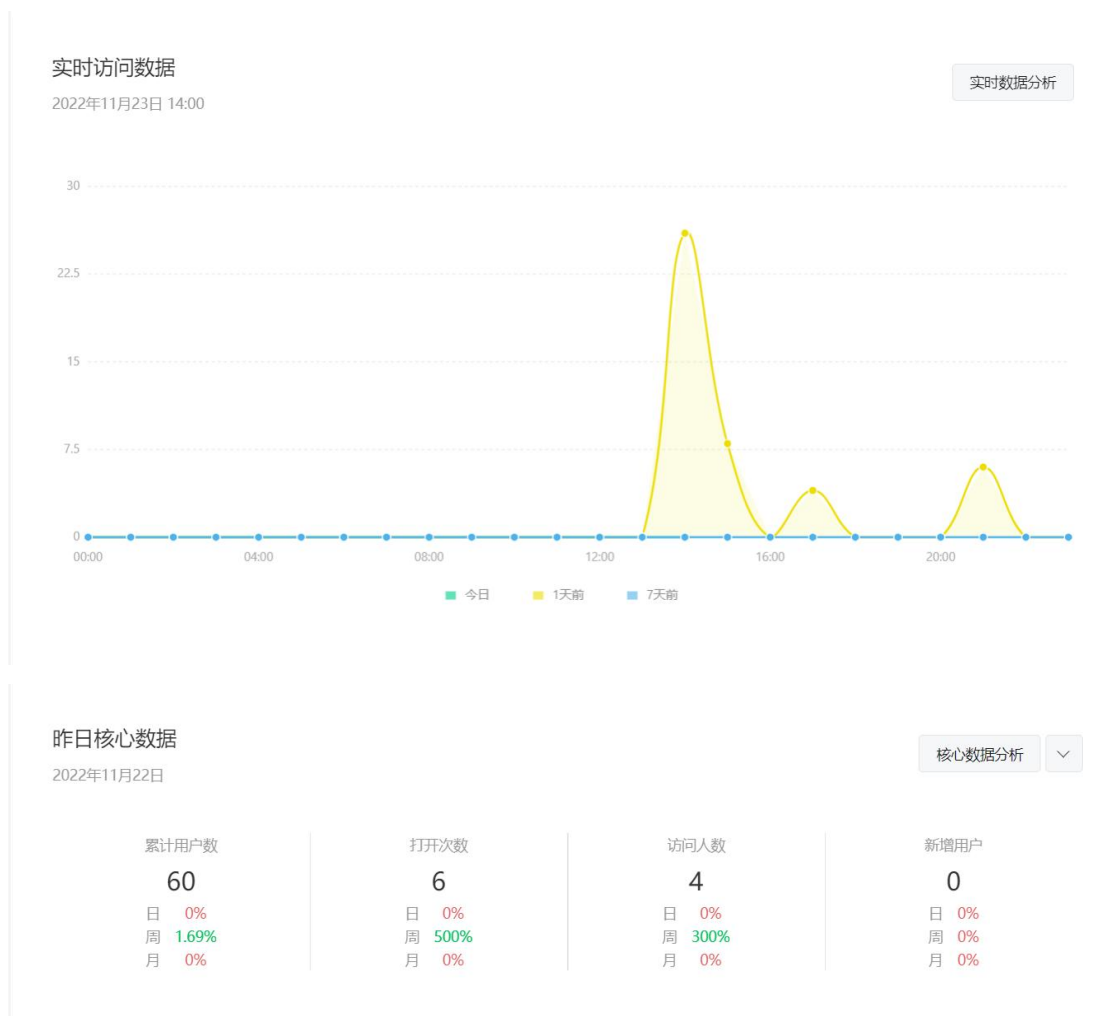


图 5.1.2 上线后注册小程序的人数和活跃用户数

如图 5.1.3，大家在使用了小程序后也对其所实现的功能和优美的界面赞不绝口，并满足了他们的需求。



图 5.1.3 好评

5.1.2 不足

1、每道题都要点提交，没有一键提交功能

如图 5.1.4，用户提出“每一道题都要点提交”这一点不合理，改进措施：添加一键提交功能，大致方式为将用户选项保存在一个数组中，点击一键提交后

和正确答案数组对比，判断正误。



图 5.1.4 用户反馈 1

2、没有打分功能

如图 5.1.5，用户提出没有打分功能，认为分数是一种仪式感，改进措施：添加打分功能，大致方式为统计正确题数除以总题数后乘 100。



图 5.1.5 用户反馈 2

3、选择答案的点击范围太小

如图 5.1.6，用户指出每次选择必须点在选项前的圆圈里，很不方便，改进措施：查找一些方法，使点击范围变为圆圈加文字。

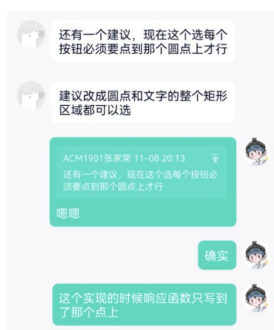


图 5.1.6 用户反馈 3

4、收藏图标点击后没有改变颜色，不美观，且同一个题目可以收藏多次

如图 5.1.7，用户指出点收藏后没有变色，不清晰明了，且多次点击会造成

多次收藏，不合理。改进措施：增加图标点击变色功能，大致方式是对图标增加点击事件 bindtap，点击后显示变色的图标，且如果这道题已经收藏了，若再次点击收藏，显示“不能重复收藏”，大致方式是设置一个变量 isCollect 记录每道题目是否被收藏，初始化为 false，在收藏图标的点击事件函数中进行判断和设置。

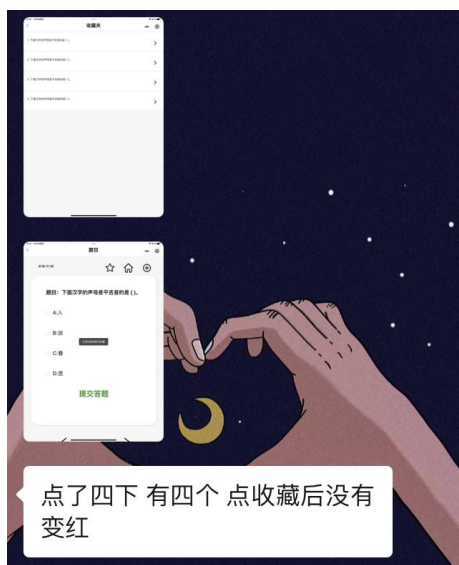


图 5.1.7 用户反馈 3

5.2 工作总结

历时一个月，小组成员基于微信开发者工具平台和 CMS 内容管理平台完成了不刷题小程序的设计开发和上线。现对主要工作做如下总结：

- （1）基于 NABCD 模型分析需求，确定了以中小學生为受众主体的刷题软件的项目目标。
- （2）基于墨刀原型设计软件，完成了以简洁美观实用为主要目标的 UI 设计
- （3）基于微信云开发平台和 CMS 内容管理平台，完成了数据模型的创建，并完善了由 word 文本生成 json 文件自动导入题库的流程。
- （4）基于微信开发者工具，完成了小程序主体的前后端设计，算法设计，实现了刷题，判断正确性、收藏题目、错题本、删除收藏和错题、生成学习记录等一系列较为完整的功能。
- （5）基于测验与反馈不断修改，实现了微信小程序开发上线。

5.3 改进方向

- (1) 增加可交流性和互动性，添加学习广场，方便用户之间分享知识点，求助问题。
- (2) 添加用使户自动上传是题目功能，使服务更加精准。
- (3) 添加计时与考试功能，调动用户积极性。
- (4) 扩充数据库，增加搜题功能。
- (5) 依据用户反馈，完善‘收藏变色’，‘一件提交’，‘计分’等功能。

6 体会

6.1 xym

分工：负责项目全过程的文档整理，分支项目的集合，总体把握项目进度。负责项目后端开发，主要包括数据库交互，算法设计。参与小程序主体设计与软件测试。

体会：在这次的软工中，我们从零实现了一个可以为用户解决问题的小程序。满满的成就感就不用说了，我也从中学到了很多东西。

首先是在小程序的相关知识上，我一开始对小程序的了解就只有平时一个普通用户对小程序的了解罢了，而在完成了这个项目之后，我对小程序有了很深刻的认识并锻炼了自己完成一个小程序的能力。在一开始确定选题的时候，我们并未将小程序作为最高优先级考虑，但是在做了一些调研之后，我们发现，在实现一个做题软件这个选题下，小程序是非常方便用户并且又很好实现的，因此我们便将其作为最终实现形式并开始做这个项目。

在一开始我们对小程序并不了解，但是我找到了微信开发者的官方文档和许多的小程序讲解视频，并将这些教程分享给了我的队友，大家一起学习，互相帮忙解决对方的问题，最后把整个教程啃完了，这也为我们打下了基础。

虽然看完了教程，但是在计算机中，理论和实战时两回事，所以我们也采用了边学边写的方法推进，首先我们先实现了一个简易的 demo 版本，完成了一些基础的选题做题功能和用户的简易界面，并在大家的共同调试下，demo 版本基本是没有问题了。于是从简到繁，我们在 demo 版本上不断添加功能和美化界面，最后实现了上线所呈现的小程序。

其次，我在团队合作中也收获了很多，在这次的合作中，我负责数据库部分，和另一位同学一起，最后与前端的同学所完成的工作进行合并。在 demo 版本中，我们首先各自实现两部分，我学习了许多关于数据库中所需要的操作，并了解了与前端的接口的实现。在 demo 版本完成之后，我们便经常开组会并交流自己的想法，大家都构思该小程序可以扩展的功能和所能优化的地方。就比如，我提出了在做题完毕时展示一个题目列表的功能，大家就通过自己所了解的知识提出了该想法的具体实现方法，最后这个扩展功能由我和另一个同学协同合作完成了，其他两位同学与此同时对界面进行了优化。整个团队对项目都很有激情并且很感

兴趣，因此大家推进起来比想象中顺利。

我从队友身上也学到了很多，在最初我是不喜欢 debug 的，看报错信息之类的事情简直就是折磨，但是我的队友非常有耐心，对报错认真分析并找到相关位置增加调试信息一步步调试，在写代码的过程中帮助了我很多，我也变得有耐心和她们一起调试代码。

最后，我也学到了其他并非完全是代码方面的知识。软件工程这门课教的不只是如何写代码，更重要的是如何写出来一个可以造福大众的软件，这不仅仅是代码质量的问题，更重要的还有 idea 的问题，如何找到用户的需求并击中用户的痛点，并对其建立可解决的模型也是很重要的，然后才是代码质量。在这门课中，我们不仅完成了代码，还对用户进行了调研去了解用户的需求和使用小程序的反馈，来不断地对小程序进行完善，最后才能实现一个可以广泛被人使用的小程序。

虽然在开始的时候我们遇到了很多问题，但是好在我们整个团队都在积极地找解决办法来解决问题并积极推进，最后才能合作完成一个可以上线的作品。

6.2 分工

分工：负责项目执行阶段成员的进度统计，UML 相关需求分析和甘特图绘制。负责数据库的设计，管理以及部分小程序后端开发，负责软件的上传与发布。

体会：本次项目我感受到了一个产品从提出构想，调研，设计，编程，测试，发布这一系列过程中的庞杂，经过这一系列的开发，我真正体会到了一个产品的完成是有多么不容易。其中，我最大的感受是认识到了在项目过程中，动态确定项目目标，和明确项目分工的必要性。但是，在大家面对一种全然未知的工具或知识时，确立明确的目标显得尤为困难，这个时候很容易不知道如何下手。我们在项目过程中也遇到过这个问题，导致了一段时间的停滞。幸好后来及时止损，重新规划当前目标，并且化大任务为子任务，重新分配，才能前进起来。另外，在面对未知的知识时，不能因为未知就退缩，而应该迎难而上，当下遇到什么问题就解决什么问题，着眼于当下，总会找到解决办法，或者，能意识到这种方法不可实现性，换一种思路。除此之外我遇到的另外一个难题是数据库的创建，由于数据库相关知识还不够扎实以及对 CMS 内容管理平台 and 微信开发工具的不了解，导致建立很多互相关联的数据库时并没有将这些关联性，独立性，高效性考

虑清晰，在写后端代码过程中又不断重新修改数据库模型，产生了很多麻烦。在制作软件发布 PPT、准备软件发布报告时，我回顾了这两个月团队成员的协作与努力。尽管我们的协作模式是稚嫩的，成员之间的协作并没有达到非常熟练而默契的程度，有时也会因观点的分歧而发生“争执”，但每位同学都是努力的，目标是统一而一致的，也正是这一股子劲儿让我们最终找到了团队协作的感觉，享受了一把“众人拾柴火焰高”的乐趣。诚然，我们并没有实现最初设计的全部功能，拍照搜题与文字搜题等，但我仍为我们能完成小程序开发以及上线感到激动，我相信，这次软件工程课程项目只是我们团队开发经历的一个开端，在课程中我学会的软件开发流程和团队协作技巧会让我受益良久。

6.3 zrk

分工：前期主要负责使用磨刀进行软件原型的制作，后期主要实现刷题界面基础代码；“刷题”界面、“收藏”界面、“错题集”界面设计与美化。

体会：《软件工程》是一门实践性极强的课程，需要丰富的软件开发知识和很强的动手能力，这是我们团队之前几乎从未接触过的领域，回顾开发过程，大致可以总结为挣扎-接受-学习-写基础功能代码-调试-增加功能完善小程序-测试-发布这几个阶段。

课程刚开始的时候，我们首先确定了与学习相关的主题——刷题，确定了用微信小程序实现，头脑中也有了应该实现哪些功能的构想，完成了 UML 图，用磨刀实现了软件原型图，但是，如何把构想实现出来是真正难住我们的地方。我之前完全没有过小程序开发的经历，对开发的流程、小程序开发工具的使用、开发语言完全不了解，网上教程众多，不知从何学起。经过我们团队的讨论和相互鼓励，我们完成了大致的分工，开始分别在网上找教程学习。我们在 b 站上看了一些小程序开发的教程，了解了小程序开发的大致流程、代码逻辑和语言规范，然后开始着手自己实现一个只有答题功能的小程序，在实现的过程中遇到问题就查看微信官方文档，查找对应的组件和 API，没有得到想要的结果时，用 `console.log` 日志输出调试，找到问题所在，最终顺利实现“不刷题”小程序雏形，但是没有数据库和界面美化。根据分工，我和bkl同学主要负责前端，即界面设计和给数据库提供接口，zym和hjl同学主要负责数据库，即微信云开发，我们就去分别学习css、微信云开发函数，最终实现了界面美化和数据库。

然后，我们加了一些拓展功能，包括用户登录与退出、错题集、收藏、删除、学习报告等，让我们的小程序更加完善，由于已经有了开发经验，实现这些功能变得更加得心应手，虽然也经过了无数次错误和调试，但是我们收获的知识 and 经验无疑是丰富的。

对我个人而言，我明白了动手实践的重要性。看教程视频的时候，光是看懂视频里的实现过程是远远不够的，必须要自己动手实现一边才能知道自己是否真正理解了；学习 css 的时候，在网上看教程十分令人头大，但是看教程的同时自己动手试一试，可以清晰地看到样式的效果，学习效果更好了，学习过程也没有那么艰难了。我深刻体悟到，很多事情看似很艰难，好像毫无头绪，这时不能望而生畏，不敢下手，要动手开始做，在做的过程中遇到问题就去解决问题，这样才有攻克困难的可能。

6.4 bkl

分工：前期主要和zrk一起负责使用磨刀进行软件原型的制作，后期主要实现刷题界面基础代码和界面的美化（“我”界面、“做题”界面、“答题结果”界面）和用户信息的数据库存储。

体会：我觉得本次软件工程课程项目是一次非常珍贵的团队协作开发软件的项目经历。从项目的立项、策划，到软件的编写、测试，再到最后的软件发布的整个过程中我逐渐认识到软件工程的复杂性与重要性，也深刻意识到团队协作的软件开发与个人软件开发的异同。

在一开始的时候，我们的进展比较顺利，在项目立项阶段，我们通过讨论很快就决定了要做一个刷题的小程序，我们把想要实现的功能全部列出，并根据我们的设想完成了 UML 图和墨刀的原型设计。但是当我们真的要开始写小程序代码的时候，我们就有一种不知道要从何入手的感觉，由于我们都没有接触过微信小程序，对微信小程序不是太了解，而且由于我们要完成的工作量很大，导致我们有一种无力的感觉。但是经过我们在网上不断的学习，我们终于开始有了一点头绪，我们把这个庞大的任务分成不同的小任务，每个人分别学习一部分的知识来完成各自的小任务，通过这个方法，我们开始写我们的小程序。

在完成本次软件工程课程项目中，我主要负责前端的界面实现和一小部分的数据库储存，在一开始写前端界面的时候由于不太会使用 css，所以导致我写前

端界面的时候需要用到很多的代码来实现我想要的效果,但后来经过对css的学习,我可以很轻易的完成一个美观的前端界面。而在写数据库储存方面,在一开始,由于不知道小程序在执行事件处理函数的时候不是是顺序执行的,所以导致我在写数据库的时候一直不能成功的把数据写进数据库,这个问题困扰了我特别久的时间,我一直以为是我哪里写错了代码,但最后发现原来是微信小程序自身的执行问题,之后我通过在网上查询,成功的找到了解决方法。

总的来说,通过这次项目开发我体会了一个小型独立游戏开发的基本流程与其面临的挑战,收获颇丰。但由于同组成员没有微信小程序的开发经验,上手这个项目花费了较长的时间。因此最后的软件还有一些原本想要实现但由于时间关系没有完成或舍弃了的功能,例如拍照搜题、用户间的交流和上传题解的功能。不过总体来说能在有限的开发时间内达到现在的这个效果让我感到十分满意。