《创新实践 2》课程报告

所在专业 计算机科学与技术

所在班级 16052317

学生学号 16041321

学生姓名 黄继升

指导老师 舒亚非

杭州电子科技大学计算机学院

2019 年 6 月

一、项目介绍

1.1 小程序项目信息

1.1.1 项目名称

《一个多功能的微信 TodoList 实现》

1.1.2 小程序名称

《hjs 的多功能 todoList》

1.1.3 小程序码

1.1.4 小程序二维码

1.2 项目背景

为什么突然想做一个 TodoList，最开始是因为实在想不出一个很好的小程序开发主题，然后平时自己又有用手机的备忘录功能来记录日常习惯和生活所思，因此就决定做一个 TodoList，另外自己在前端的技术能力上也比较欠缺，因此完成这样一个小程序对我来说既不会达到很难的程度，又可以加深我对前端知识的掌握。关于如何去实现好一个 TodoList，我想一个 TodoList 应不仅应该具有用户的日常管理、事项记录功能，如果能够在更大程度上方便用户的操作，并且能够和别人分享自己的日常，并且加点娱乐化的功能，那受众面应该也会比较广。

1.3 开发环境及使用框架

（1）微信小程序开发工具 & vscode

（2）微信小程序开发基本框架、云开发框架

（3）腾讯云开发数据库、云存储

（4）有赞前端 UI 框架的引入和使用

1.4 项目学习目标:

（1）了解微信小程序的概念及前景发展；掌握微信小程序开发工具的使用；掌握微信小程序的基本开发流程以及开发规范，体验微信的云开发功能

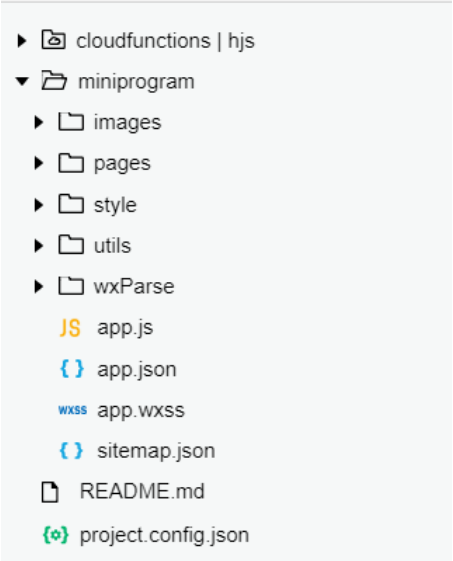
（2）系统全面地对前端知识的基础知识进行查漏补遗，熟悉 Js 基本语法，为接下来学习 vue.js、React.js 等前端框架做好铺垫。

（3）初步了解 Js 在后端开发上的用法：node.js

（4）掌握 git 的一些常用操作和基本命令，比如 clone、add、commit、push、pull 等这些非常基本常用到的命令，学习如何利用 github 发掘好的项目进行学习，并且学会用 github 进行团队的协同开发。

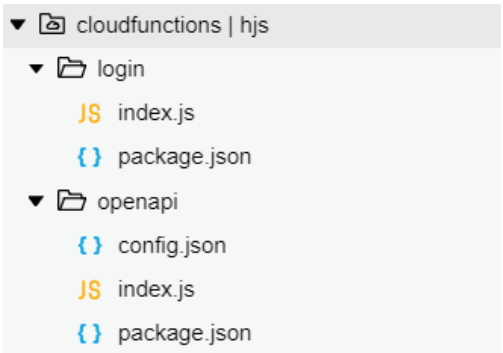
二、项目结构和技术架构

2.1 小程序项目结构



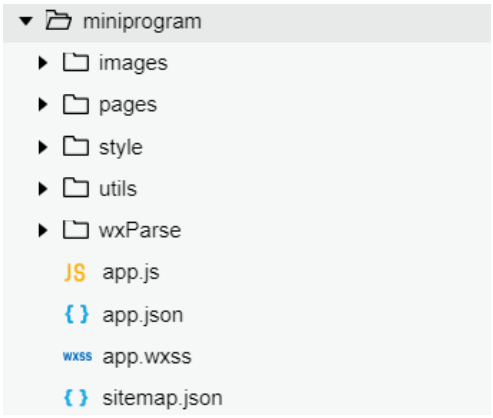
2.1.1 cloudfunctions|hjs 目录

小程序的云开发环境，因为我的项目没有自编后端，因此需要用到云数据库和云存储作为后端存取功能。其中默认定义了一些基本的云函数。这次项目中我没有定义自己的云函数。



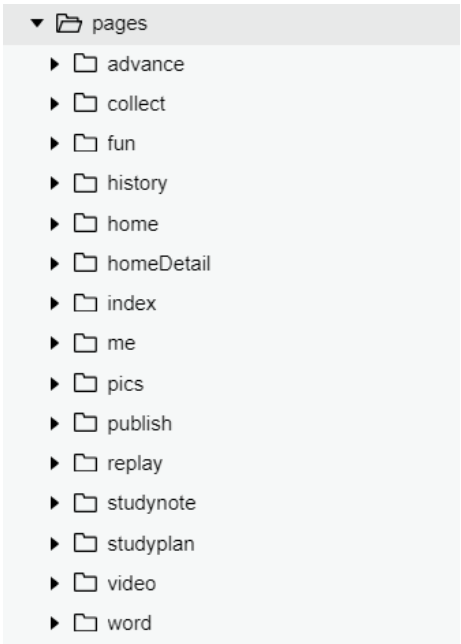
2.1.2 miniprogram 目录

项目小程序的主要代码目录，其中有文件夹 images、pages、style、utils、wxParse，以及整个项目全局内容 app.js、app.json、app.wxss,另外还有适配微信搜索的 sitemap.json。目前我还没对这个文件进行研究过，好像是说 sitemap 功能可以默认收录所有小程序的页面内容，用于微信搜索场景。



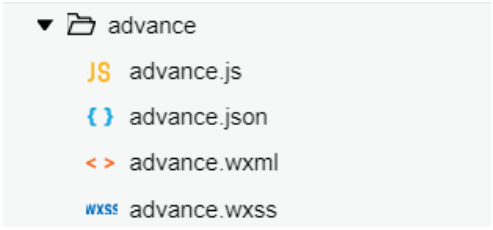
（1）images 文件夹：存放整个小程序需要用到的所有图片资源

（2）pages 文件夹：存放小程序所有的页面代码，包括 wxml、wxss 以及 js 和 json 文件。

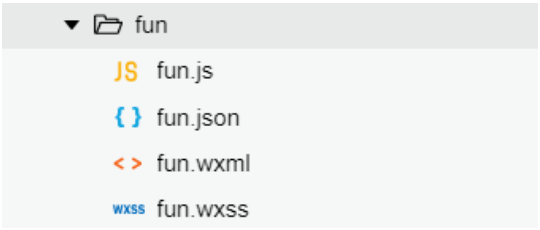


pages/advance 文件夹:定义小程序的用户意见反馈功能

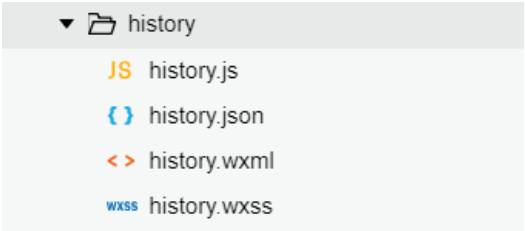
pages/advance 文件夹:定义小程序的用户意见反馈功能



pages/fun 文件夹: 定义小程序的娱乐功能导航界面

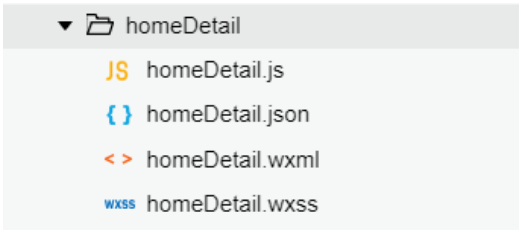


pages/history 文件夹: 定义小程序的查询动态发布历史功能



pages/home 文件夹：定义小程序的社区动态界面功能

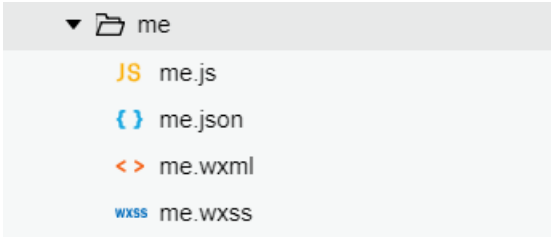
pages/homeDetail 文件夹：定义小程序的每条社区动态的详细内容界面



pages/index 文件夹：定义小程序的每条社区动态的详细内容界面

pages/me 文件夹：定义小程序的用户中心界面

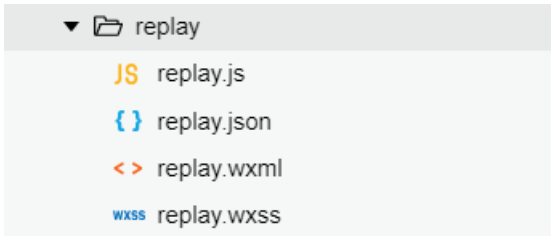
pages/me 文件夹：定义小程序的用户中心界面



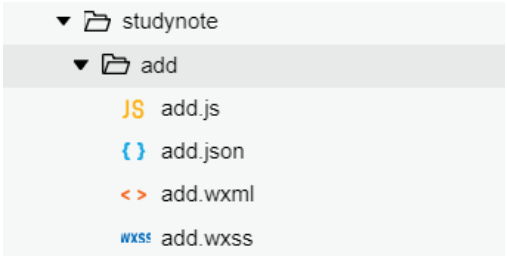
pages/pics 文件夹：定义小程序的娱乐功能——搞笑动态图浏览

pages/publish 文件夹：定义小程序的发布社区动态功能

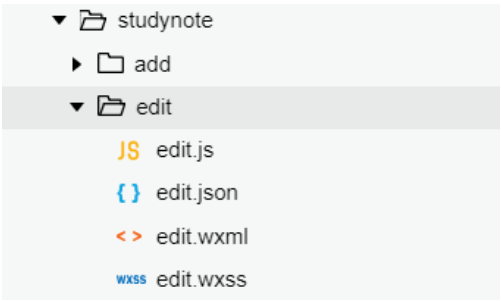
pages/replay 文件夹：定义小程序的社区动态评论功能



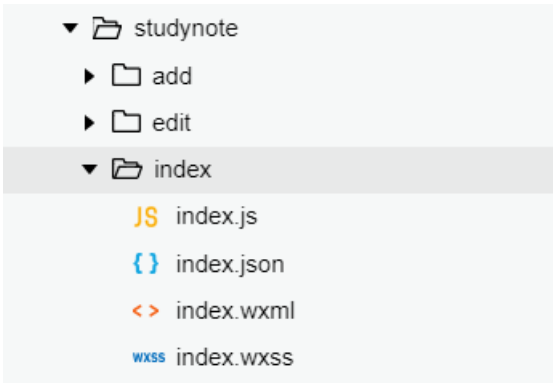
pages/studynote/add 文件夹：定义小程序添加笔记功能



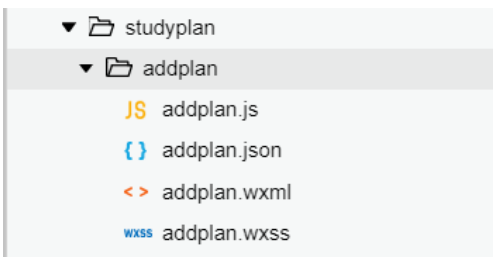
pages/studynote/edit 文件夹：定义小程序修改笔记功能



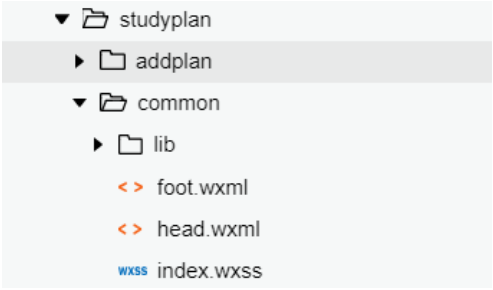
pages/studynote/index 文件夹：定义小程序笔记功能主界面



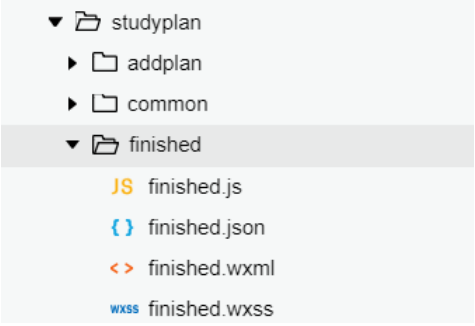
pages/studyplan/addplan 文件夹：定义小程序添加时间计划功能



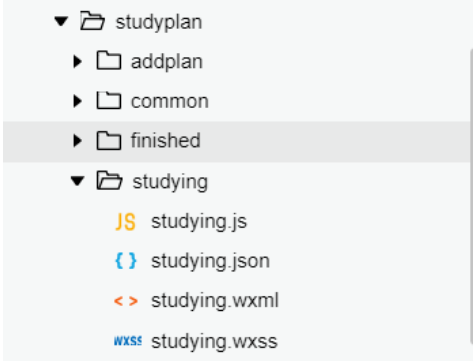
pages/studynote/common 文件夹：定义小程序时间计划功能的界面模板。



pages/studynote/finished 文件夹：定义小程序的完成计划列表界面

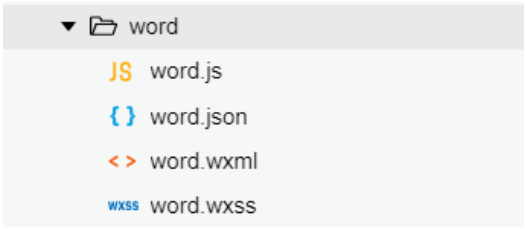


pages/studynote/studying 文件夹：定义小程序未完成计划列表界面



pages/video 文件夹：定义小程序的娱乐功能——有趣视频浏览

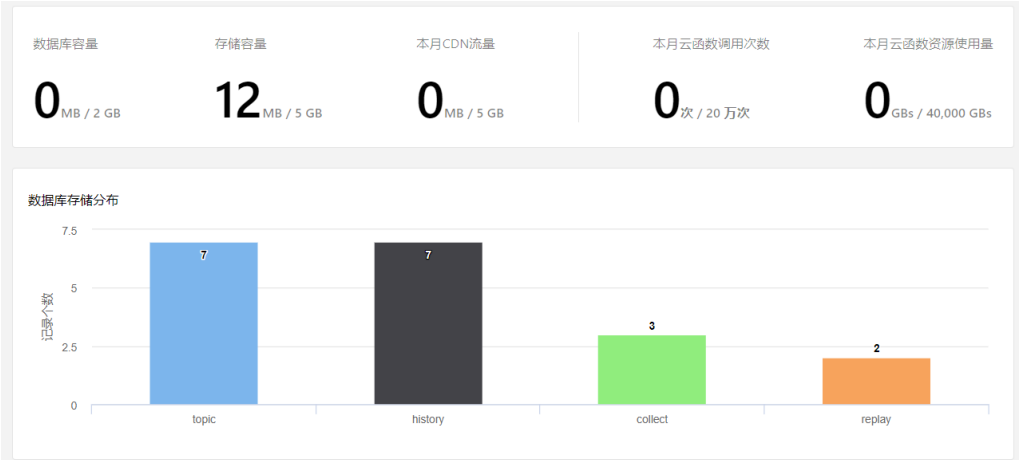
pages/word 文件夹：定义小程序的娱乐功能——搞笑段子浏览



（3）utils 文件夹：存放一些工具类的函数操作，如对日期时间的处理函数，获取数组、对象的长度函数以及分享小程序时的随机生成语录等等。

（4）wxParse 文件夹：这是微信的富文本解析工具，之前引入是打算用来学习一下如何解析 html 文本的，后来由于时间的关系没有用到。

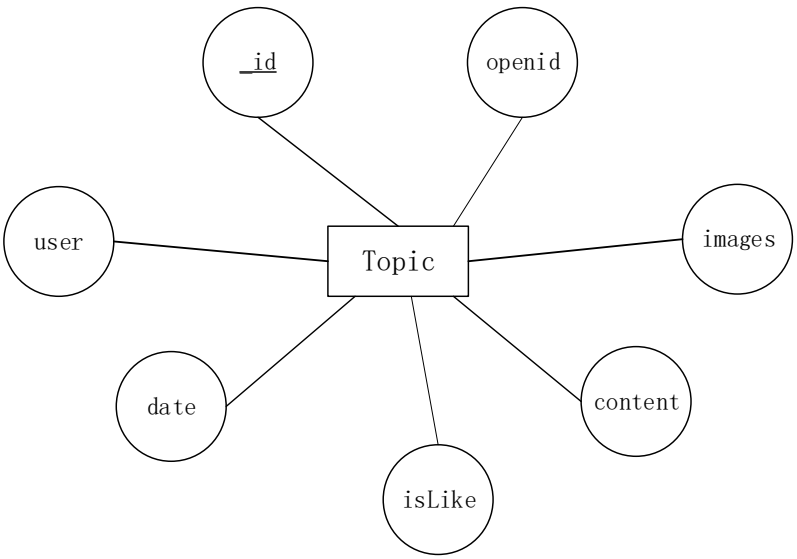
2.2 云开发数据库表结构



云数据库中定义了四个表，分别为 topic 表、history 表、collect 表以及 replay 表

（1）topic 表：

ER 图：



主键：_id（唯一标识所有用户所发过的每一条动态）；

其他属性：openid（发布用户的 openid），

content（动态文本内容），

date（动态发布日期），

images（该动态的图片在云服务器上的存储位置），

isLike（是否曾被当前用户收藏），

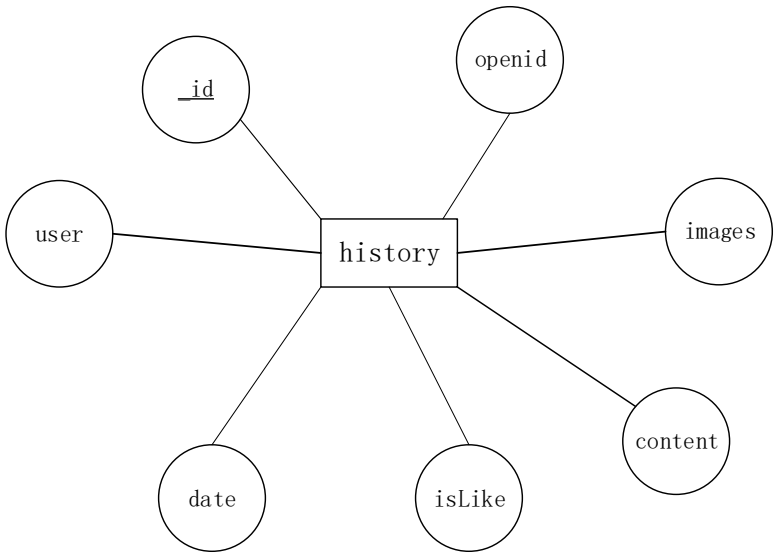
user（发布用户的详细信息，包括头像、昵称、地址等等）

eg:


```
"_id": "2d9d2d8c5cfc62070238e3b85d8ab9fa"
"_openid": "oLM575X8DAwc6rxmZMOJ4rPxbMdM"
"content": "分享几段微信小程序页面跳转代码"
"date": Sun Jun 09 2019 09:33:59 GMT+0800 (中国标准时间)
▸ "images": ["cloud://hjs-vfa14.686a-hjs-vfa14-1258933147/4309_346.p...]
"isLike": false
▸ "user": {"avatarUrl":"https://wx.qlogo.cn/mmopen/vi_32/iciaDLDYR...}
```


（2）history 表：

ER 图：



主键：_id（唯一标识每一条当前用户的发送历史动态）；

其他属性：openid（发布用户的 openid），

content（动态文本内容），

date（动态发布日期），

images（该动态的图片在云服务器上的存储位置），

isLike（是否曾被当前用户收藏），

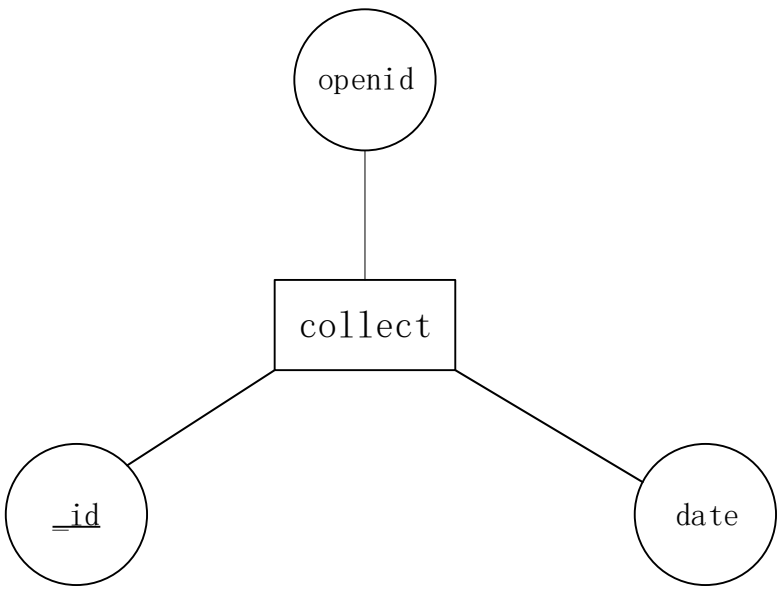
user（发布用户的详细信息，包括头像、昵称、地址等等）

eg:


```
"_id": "dec80a9e5cfc6372023775a552c57d67"
"_openid": "oLM575X8DAwc6rxmZMOJ4rPxbMdM"
"content": "Go语言，从入门到放弃"
"date": Sun Jun 09 2019 09:40:02 GMT+0800 (中国标准时间)
▸ "images": ["cloud://hjs-vfa14.686a-hjs-vfa14-1258933147/8334_732.p...]
"isLike": false
▸ "user": {"avatarUrl":"https://wx.qlogo.cn/mmopen/vi_32/iciaDLDYR...}
```


（3）collect 表：

ER 图：



主键：_id（唯一标识每一条当前用户收藏的动态）；

其他属性：openid（发布用户的 openid），

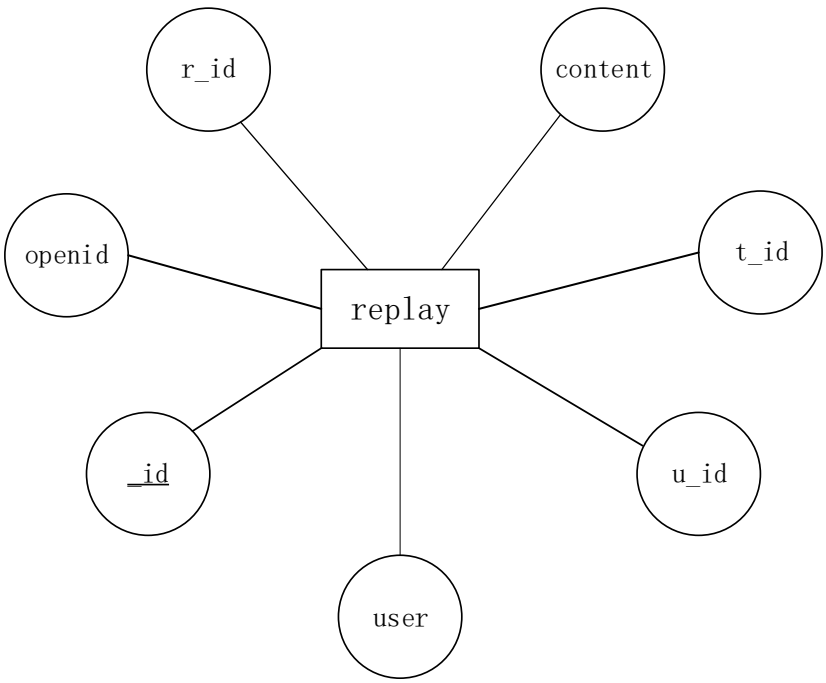
date（动态发布日期），

eg:


```
"_id": "2d9d2d8c5cfc62070238e3b85d8ab9fa"
"_openid": "oLM575X8DAwc6rxmZMOJ4rPxbMdM"
"date": Sun Jun 09 2019 09:34:11 GMT+0800 (中国标准时间)
```


（4）replay 表：

ER 图：



主键：_id（唯一标识每一条动态的回复 id）；

其他属性：openid（发布用户的 openid），

content（动态文本内容），

r_id（动态发布日期），

t_id（该动态的图片在云服务器上的存储位置），

u_id（是否曾被当前用户收藏），

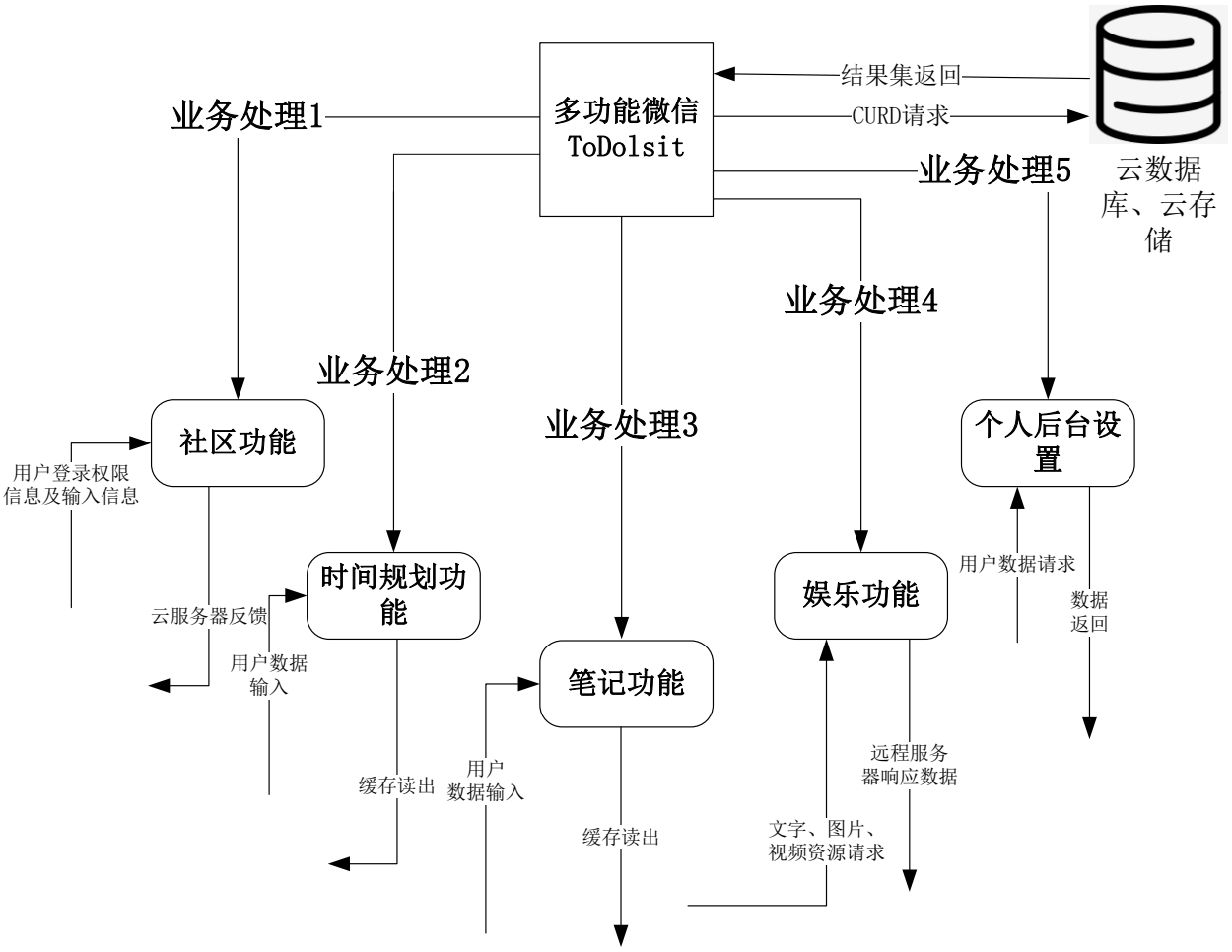
user（发布用户的详细信息，包括头像、昵称、地址等等）

eg:


```
"_id": "cbdb4c165cfc632a0238c2d031d1bdf5"
"_openid": "oLM575X8DAwc6rxmZMOJ4rPxbMdM"
"content": "Helloworld"
"date": Sun Jun 09 2019 09:38:49 GMT+0800 (中国标准时间)
"r_id": "2d9d2d8c5cfc62070238e3b85d8ab9fa"
"t_id": "2d9d2d8c5cfc62070238e3b85d8ab9fa"
"u_id": "undefined"
▸ "user": {"avatarUrl":"https://wx.qlogo.cn/mmopen/vi_32/iciaDLDYR...}
```

2.3 技术架构和关键技术点

2.3.1 技术架构图



2.3.2 关键技术难点

（1）关键点 1：

在社区功能的发布和评论功能中，必须确定用户是否处于登录状态判断来进行检测和判断，如果用户之前已经授权登陆了，则成功发布社区动态并跳转到社区动态页面；如果用户没有授权登录，则无法进行发布社区和进行评论，并且原有发布区或评论区填写的内容会被清空。如何实现整个逻辑看似很简单，但是其中需要注意的点会很多。对此我自己构思了两个完整的解决思路。

解决思路 1：

在 publish.js 定义一个布尔变量 flag，初始值为 false，定义判断用户是否登录的状态函数 judgeUserLogin()，当用户点击进入发布页面时，页面加载会自动触发 onLoad 方法，并在 onLoad 方法里调用 judgeUserLogin()，如果用户已授权登录，那么 judgeUserLogin()将会把 flag 赋值为 true。当用户完成输入文字和图片时，点击发布，会调用 formSubmit 方法，formSubmit 方法内对 flag 进行判断，如果 flag = true，则成功提交到服务器并跳转到社区动态页面查看更新，此时用户可以查看更新。如果 flag = false，则直接调用授权登录函数，来获得授权弹窗，此时用户点击同意，则成功授权，成功提交并跳转到社区动态页面查看更新;如果用户点击拒绝授权,则回到原来的页面。

解决思路 2：

这个解决思路比较简单，也不需要对用户状态进行初始化的判断，只是需要每次当用户点击发布时，就调用 wx.getSetting()这个异步方法，如果已经授权登录，则通过内部的回调函数直接将数据发送到服务器成功提交，并跳转到社区动态页面查看更新；如果之前没有授权登陆过，则会出现授权弹窗要求用户点击同意授权，然后用户点击同意后，需要重新点击一次发布按钮，才能通过成功回调提交到服务器，并跳转到社区动态页面查看更新。整个逻辑非常简单，但是可能操作上会比较费一点点时，因为每次都需要调用 wx.getSetting()这个函数，毕竟调用函数会比直接判断 flag 比较费时一点，但是效果上没有差别。

最终实现方案：

一开始使用的是解决思路 1，但是我在实现的时候还是遇到很多问题，而且到现在还是没有解决，先给出我用思路 1 写的 formSubmit()函数逻辑代码：

```
formSubmit: function (e) {
  var p = new Promise(function (resolve, reject) {
    //无法实现，that.jugdeUserLogin()在执行时，resolve 仍然通知
    setTimeout(function () {
      that.jugdeUserLogin();
      resolve();
```

```
      }, 0)
    });
    p.then(function(resolve){
     that.data.content = e.detail.value['input-content'];
     if (flag) {
      if (that.data.canIUse) {
        if (that.data.images.length > 0) {
          that.saveDataToServer();
        } else if (that.data.content.trim() != '') {
          that.saveDataToServer();
        } else {
          wx.showToast({
            icon: 'none',
            title: '写点东西吧',
          })
        }
      } else {
        that.jugdeUserLogin();
        console.log("出错");
      }
    }
    else {
      return;
    }
    })
}
```

我原先料想用 promise 来实现验证和提交的先后同步提交操作，结果当用户之前没有授权登录时，点击发布，会弹出授权窗口让用户授权，但是此时数据和用户状态却已经被提交到了服务器，然后直接跳转到了社区动态页面，此时虽然数据提交了，但是用户头像和用户名却仍然为空。后经过代码的逐步调试，发现当 that.jugdeUserLogin()在执行时，也就是还未更新 flag 时，resolve 仍然通知异步操作已经完成并执行成功回调。而 jugdeUserLogin()函数是我对 wx.getSetting()进行进一步的封装，这也是一个异步方法。所以应该是异步方法里面不能再使用异步方法。我最终使用的是解决思路 2 来实现这个逻辑，给出关键代码：

1. wxml 页面使用 form 元件作为框架；

```
<form bindsubmit="formSubmit">
 …
</form>
```

2. form 内部使用微信提供的按钮元件，该元件可实现弹窗授权

```
    <button wx:if="{{canIUse}}"class="btn" formType='submit'
open-type="getUserInfo">发布广场</button>
```

3. publish.js 实现 formSubmit()函数

```
formSubmit: function (e) {
 wx.getSetting({
   success: function (res) {
     if (res.authSetting['scope.userInfo']) {
       // 已经授权，可以直接调用 getUserInfo 获取头像昵称
       wx.getUserInfo({
         success: function (res) {
           that.data.user = res.userInfo;
           that.data.content = e.detail.value['input-content'];
           if (that.data.canIUse) {
             if (that.data.images.length > 0) {
               that.saveDataToServer();
             } else if (that.data.content.trim() != '') {
               that.saveDataToServer();
             } else {
               wx.showToast({
                 icon: 'none',
                 title: '写点东西吧',
               })
             }
           } else {
             that.jugdeUserLogin();
             console.log("出错");
           }
         }
       })

     }
     else {
       console.log("没有授权登录");
       return;
     }
   }
 })
}
```

参考链接： https://www.cnblogs.com/sweeeper/p/8442613.html

(2) 关键点 2：手势底部上拉加载数据

出现问题：

这是我第一次写手势操作的函数，而且结合了第一次使用 api 接口来请求图片数据，所以一开始也比较迷惑，到底应该怎样组织代码逻辑才能达到自己想要的效果，是一个问题。所以也在此记录。

解决思路 1：

这是自己想的思路，使用微信元件 scroll-view，并利用其属性 bindscrolltolower 定义一个下拉事件监听函数 lower，lower 内部调用函数 fetchJoke，fetJoke 函数逻辑如下：

当触发下拉函数 lower 后，执行 fetJoke 函数，根据 api 提供的请求参数规则来请求图片数据，得到下一页的图片数组，然后将该数组与页面原来的数组用 concat 连接到末尾，最后用 setData 来更新页面数据，就可以实现这个逻辑了。

解决思路 2：

直接利用微信的 onReachBottom()函数，不使用其他元件，然后直接在这个函数中执行和实现思路 1 中的逻辑。这里不再做相同的阐述。

最终解决方案：

采用解决思路 1，在页面上拉触底时间的处理函数编写逻辑代码如下：

```
fetchJoke: function () {
 wx.showNavigationBarLoading();
 var that = this;
 wx.request({
   url: requestUrl,
   data: {
     'showapi_appid': app.globalData.appid,
     'showapi_sign': app.globalData.apiKey,
     'page': curPage.toString(),
     'type': app.globalData.tImg
   },
   method: 'GET',
   success: function (res) {
     // success
```

```
            console.log("success");
            if (curPage == 1)
              that.setData({ jokes:
res.data.showapi_res_body.pagebean.contentlist });
            else
              that.setData({ jokes:
that.data.jokes.concat(res.data.showapi_res_body.pagebean.contentlist)
});

            curPage = curPage + 1;
            wx.hideNavigationBarLoading();
            if (isPullDownRefreshing)
              wx.stopPullDownRefresh();
          },
          fail: function () {
            // fail
          },
          complete: function () {
            // complete
          }
        })
      }
```

参考链接：https://www.jianshu.com/p/da4e6dfe4ffd

四、小程序效果展示

图 4-1 小程序主界面

图 4-2 笔记功能主界面（记录为空）

图 4-3 缓存中有记录时的笔记界面

图 4-4 笔记添加编辑界面

图 4-5 保存后的笔记界面

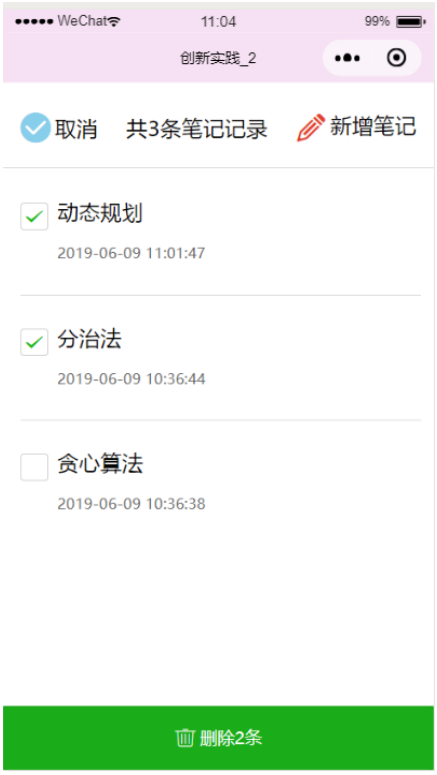


图 4-6 选中删除界面

图 4-7 时间规划表界面

图 4-8 添加规划界面

图 4-9 保存后的界面显示，以倒计时的方式呈现

图 4-10 选中某一项记录后，可选择完成或放弃

图 4-11 已完成任务栏详情

图 4-12 社区功能，查看用户最新动态（支持多图上传）

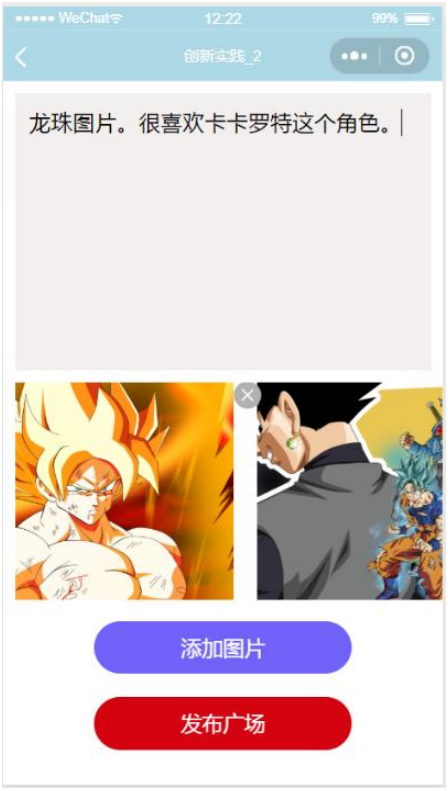


图 4-13 上传用户动态界面（支持多图上传）

图 4-14 成功发布动态

图 4-15 点击进入动态详情界面，可以点赞，也可以回复。

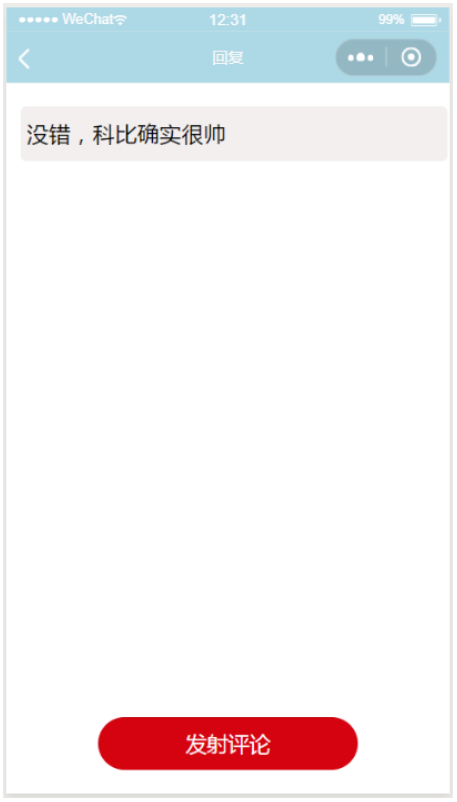


图 4-16 撰写评论，点击发送

图 4-17 发送成功后跳转到动态详情页可看到评论效果

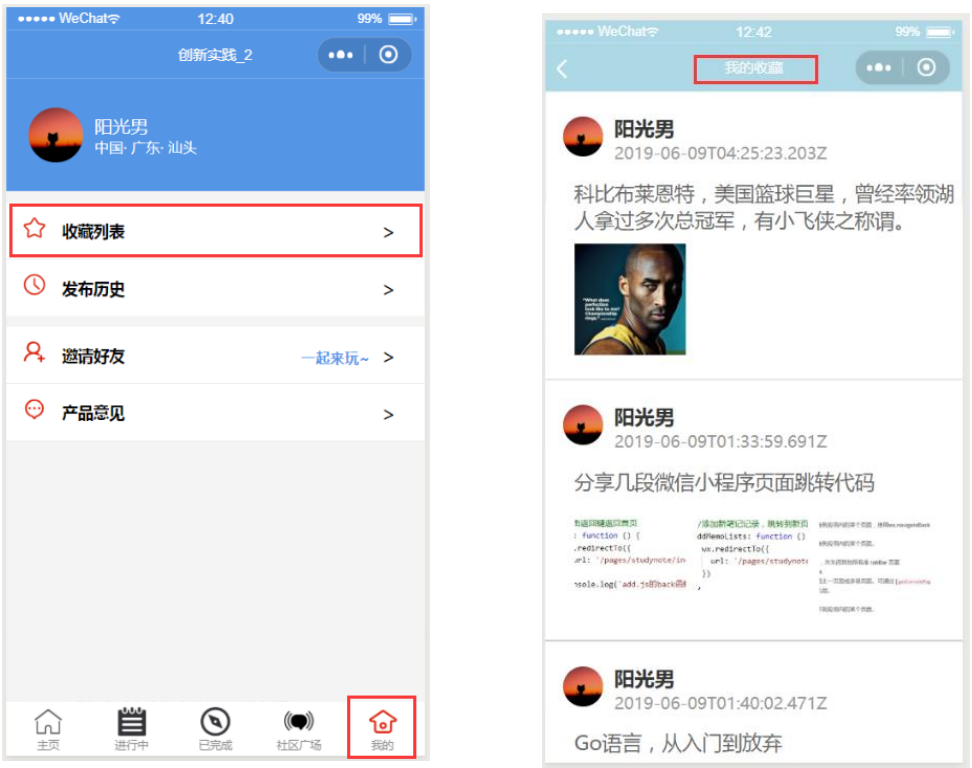


图 4-18 在“我的”->“收藏列表”可以看到用户所有点赞收藏的动态

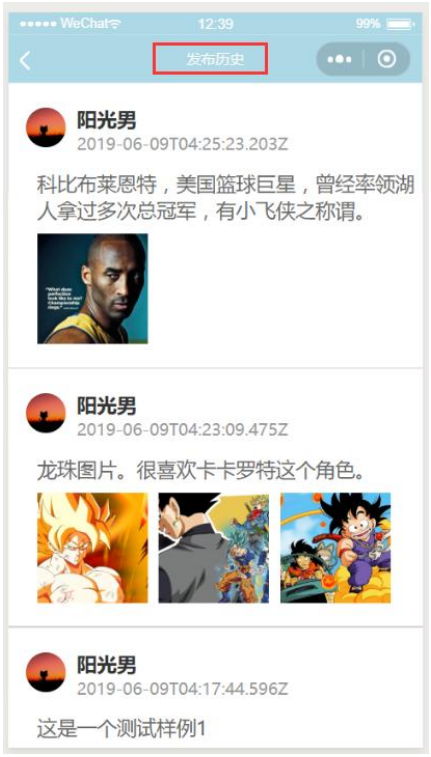


图 4-19 在“我的”->“发布历史”可以看到自己所有历史动态

图 4-20 休闲娱乐功能

图 4-20 有趣动图&精彩视频，休闲娱乐

四、自我评分细则

4.1 作品评分表

考查点	占分	自评	自己的理由
基本 CRUD	20	20	基本增、删、改、查都有。既有微信缓存的 CRUD，又有云数据库和云存储的 CRUD
后端数据存取(云开发或自编后端均可)	10	10	由于是单人项目，而且从来没使用过云开发，所以这次项目采用了云开发模式，实现起来也比较便捷简单
UI 设计与样式实现	10	7	我个人平时学得后端比较多点，前端学得不好。这次小程序 UI 设计确实一般，用户体验一般
导航(全局导航栏、tabbar、分类、分页等)	10	10	该有的全局导航栏、tabbar、分类都有了。基本符合要求
基于手势的操作	10	10	在社区功能中，用户获取更多的动态内容、以及娱乐功能中的下拉获取更多视频和图片功能，都用到了手势操作
登录与会话	10	10	简单的用户授权验证，如果用户拒绝授权，将无法进入社区进行动态发布和评论功能
多媒体相关	10	8	社区的发布动态功能支持拍照上传图片，这里使用到了照相机
代码模块化	10	5	第一次学习小程序，不是很理解代码模块化。仍有很多不足之处
代码编写规范	10	5	基本能够遵循代码编写规范，但是仍然存在着很多不足之处。

总分	100	85	综合了自己的项目得分点，再最后经过验收后老师给出的评价来给定分数

4.2 总分表

考核项目	考核内容	考核依据与方法	占总评成绩的比重	得分
实践作品	作品质量	根据学生作品的完成度、难易程度、创新性等评分	40%	85
团队表现	出勤率、交互情况、是否按进度完成	根据学生出勤率，与组员、导师的交流沟通情况、按进度完成情况等评分	20%	85
项目答辩	答辩质量	按学生答辩的材料准备、答辩质量、互动情况等评分	20%	85
项目文档	文档质量	按文档规范性、各部分内容质量，批阅评分	20%	85
小计			100%	85

五、项目总结

这学期学习开发小程序，真的收获巨大。首先作为一个立志于做后端开发的学生来说，熟练掌握一些前端开发技能也是必不可少的，而且我上学期主要是以学习后端为主，对前端的学习有些忽略，这学期也算是把前端的基础知识全部都掌握下来了，接下来的任务就是进阶前端框架，以及继续学习和深化我的 Java、go 后端技能储备库。实际上，我之前也做过一些手机移动开发的项目，比如安卓 App 等，所以我觉得小程序的技术架构是融合了移动 App 开发和前端开发的特点，它相比于移动 App 开发过程的繁琐和体系的庞大，显得更为轻量和便携，而且占用系统资源少，开发历程迅速简短，一下子我就喜欢上它了。这次项目开发美中不足的就是开发的过程中一直对项目的客户和功能定位不明确，所以功能虽然还是很多的，但是整个项目并不能很好地贴合用户的需求，而且给人一种强加功能的感觉，也没能完成老师让我完成的共享 toDolist 的功能，但是我觉得整个过程下来我还是尽我所能，并且从中也收获巨大，整个项目跑在微信环境中运行时，效果还是很不错的，也算是不负自己的努力和心血吧。

最后特别感谢舒亚非老师对我本学期项目开发的指导，在整个开发和学习中老师对我的帮助是最为关键的，也谢谢创新实践班其他同学给出的宝贵意见和评价。