**“书来书往”小程序**

实

现

分

析

报

告



G21（陈昱熹，马宇轩，王敏）

目录

一．引言 3

1.1编写目的 3

1.2项目背景 3

1.3参考资料 3

二．“书来书往”小程序设计说明 4

2.1程序清单 4

2.2小组代码规范 4

三．项目计划展示 7

四．界面原型展示 8

五．软件测试 9

5.1测试计划 10

5.2测试准备 11

5.3单元测试 14

5.4集成测试 15

5.5确认测试 15

5.6白盒测试 15

# 引言

## 1.1编写目的

本次实现测试分析是在小程序部分未完全成型的情况下依据课程实现阶段评审表进行的，针对现有功能和界面的实现进行测试，测试结果并不完善完整，后期将会继续测试、更新。

## 1.2项目背景

被测试的软件项目名称：“书来书往”小程序；

项目的任务提出者：本课程老师及本开发小组组长；

项目的任务开发者：本项目开发小组G21；

项目的用户：浙江大学城市学院广大师生群体，以学生群体为主；

本文档的阅读者：本项目开发小组成员及项目经理（本课程教师）。

## 1.3参考资料

1、微信公众平台《小程序开发指南》

2、软件工程导论[M]. 清华大学出版社, 张海藩.牟永敏, 2013

3、软件工程-实现阶段评审表

4、小程序代码书写规范 https://blog.csdn.net/qq\_33237207/article/details/88943455

https://www.jianshu.com/p/79814d30708f

# 二．“书来书往”小程序设计说明

## 2.1现有程序清单

见附件。

## 2.2小组代码规范

**.css规范**  
小程序在开发过程中布局使用flex布局，单位用rpx字体使用px，css命名用 - 作为单词分隔符。

css命名示例

v-image{

disaplay: flex;

flex-dirextion: row

width: 100rpx;

font-size: 14px;

}

. **js规范**  
js命名用驼峰命名,JS语句无需以分号结束，统一省略分号JS中一致使用反引号 ``或单引号' ' , 不使用双引号。WXML、CSS、JSON中均应使用双引号。点击事件规范,事件函数命名方式为 on + 事件名或者业务名

js命名示例

async onGetList() {

let imageContent = 'js中使用单引号'

.....

}

props传入值，后面如果需要修改的化，再data中重新命名变量前面添加 \_ 下划线作为区分

props = {

list: {

type: Object,

default: []

}

};

data={

\_list:[],

}

回调函数统一使用Promise函数的方式进行编写，回调成功的参数统一为res，错误参数为err

// promise 处理回调

let back = new Promise((resolve, reject) => {

if (/\* 异步操作成功 \*/){

resolve(value);

} else {

reject(error);

}

});

back.then((res) => {

console.log('成功回调！', res);

}).catch((err) => {

console.log('失败回调！', error);

});

链接：https://www.jianshu.com/p/79814d30708f

**通用规范**

\* 尽量使用完整的英文描述符

\* 采用适用于相关领域的术语

\* 采用大小写混合使名字可读

\* 尽量少用缩写，但如果用了，要明智地使用，且在整个工程中统一

\* 避免使用长的名字（小于 15 个字母是个好主意）

\* 避免使用类似的名字，或者仅仅是大小写不同的名字

\* 避免使用下划线（除静态常量等）

\* 注释应该增加代码的清晰度

\* 保持注释的简洁

\* 在写代码之前写注释

**文件名规范**

pages目录下文件名用英文单词命名，多个单词用驼峰命名法，以小写字母开头，尽量不包含数字或者其他字符。

**.wxml文件---页面代码规范**

A、编码均遵循`XHTML`标准,标签、属性、id命名由小写英文、数字和‘\_’组成，且所有标签必须闭合，属性值必须用双引号`""`。

B、避免使用中文拼音尽量简易并要求语义化。

C、尽可能减少标签嵌套，做到代码层级清晰。

D、尽量避免在标签上直接写样式。

**.wxss文件---页面代码规范**

A、使用选择器时，命名比较短的词汇或者缩写的不允许直接定义样式。如下：

.hd,.bd,.td{};

可用上级节点进行限定。如下：

.recommend-mod .hd{ }

B、多选择器规则之间建议换行，即当样式针对多个选择器时每个选择器占一行。如下：

button.btn,

input.btn,

input[type="button"] {…};

C、使用z-index属性尽量z-index的值不要超过150（通用组的除外），页面中的元素内容的z-index不能超过10，（提示框等模块除外但维持在150以下），不允许直接使用（999~9999）之间大值。

D、避免使用低效的选择器。如下：

body > \* {…};

ul > li > a {…};

#footer > h3 {…};

**.js文件---页面代码规范**

A、函数名前必须注释此函数的参数意义，以及该函数的用途。

B、if、while、for、do语句的执行体用"{}"括起来，"{}"格式如下：

if (a==1) {

//代码

};

C、避免额外的逗号。如下：

var arr = [1,2,3,];

D、长语句可考虑断行，即一行显示不下，可分行显示。

E、使用严格的条件判断符。用===代替==，用!==代替!=，避免掉入==造成的陷阱 。

在==时，则会有一些让人难以理解的陷阱：

<script>

var undefined;

console.log(undefined == null); // true

console.log(1 == true); //true

console.log(2 == true); // false

console.log(0 == false); // true

console.log(0 == ''); // true

console.log(NaN == NaN);// false

console.log([] == false); // true

console.log([] == ![]); // true

</script>

F、下面类型的对象不建议用new构造。

new Number

new String

new Boolean

new Object //用{}代替

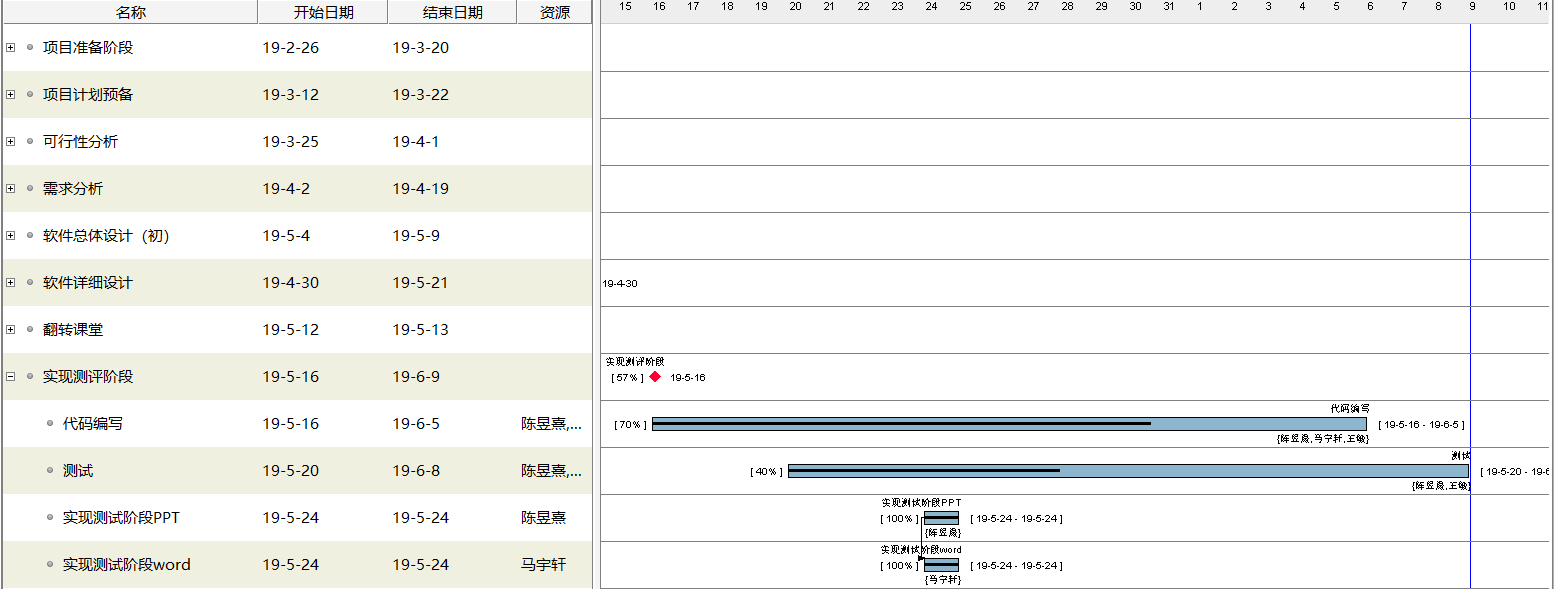
new Array //用[]代替

**图片规范**

命名应用小写英文、数字、\_组合，便于团队其他成员理解。

图片格式仅限于gif、png、jpg等，单张图片大小最好不要超过100K。

# 项目计划展示



# 四．实现展示

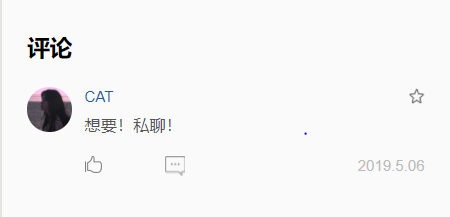
主界面



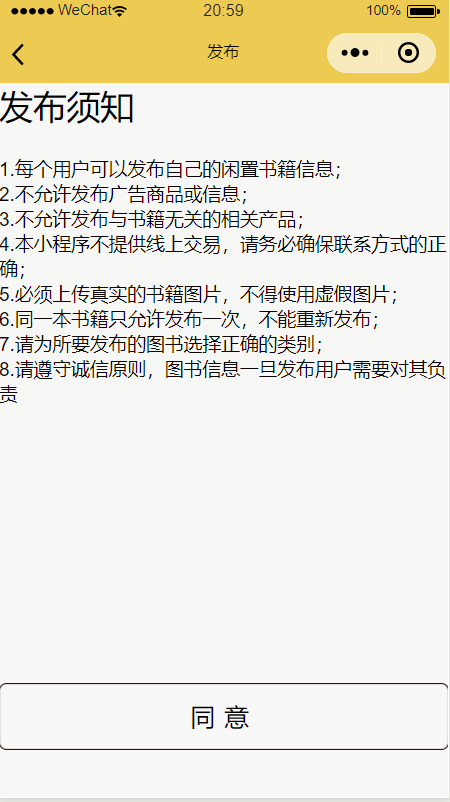
详情页

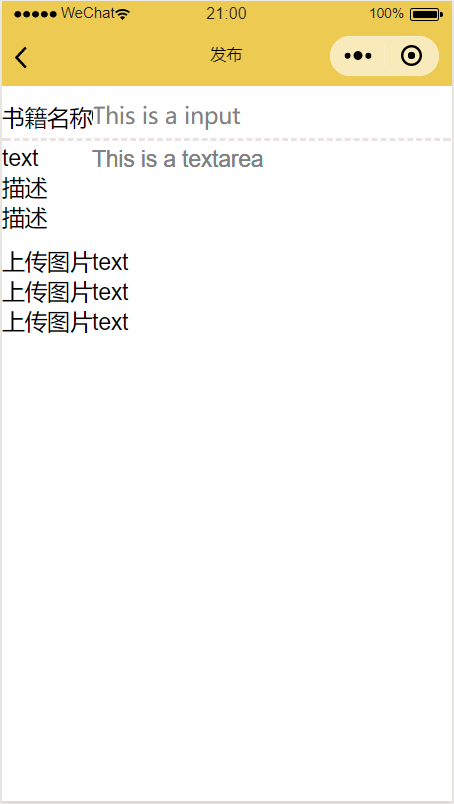


评论



发布页面





个人页面



# 五．软件测试

## 5.1测试计划

**1.权限测试**

需要检查以下几种情况下微信用户访问的权限  
1）未授权微信登录小程序  
未授权时，一般使用一些业务功能的时候，都会弹出提醒：先授权再操作对应功能。or在提交数据到后台的时候，会提示补充相关身份信息才能提交成功  
2）已授权微信登录小程序  
授权微信访问小程序，意味着自己的微信账号可被小程序管理方所获取，自动以微信的身份行使业务操作权限，比如咨询、支付、数据查询等  
3）同一微信号在不同手机端登录授权查看数据权限  
同一微信号在不同手机微信端授权登录同一小程序之后，所能查看的数据和操作的权限都应该是同步一致的

**2.功能测试**

1）按功能模块测试  
根据设计好的各个大类功能模块划分，然后再逐级细化，覆盖到每个功能尽可能全面的测试点  
2）按业务流程测试  
小程序的业务，比如咨询、支付、播放、查询、下载。把各个功能点串联起来形成完整的业务流程来检查；同一个业务，可能有不能的路径来实现，每个路径都需要覆盖检查  
3）按数据流向测试  
根据数据从某一端操作输入和输出流向，设计基于数据流的测试用例，输出的数据也可能成为另外一端的输入，检查输入的数据是否按照代码逻辑执行正确的输出，是否数据发生异常（无法输入；有输入却无任何输出；输出不正确；多余的输出其他信息...）  
4）同一功能不同的入口有效性的检查  
小程序中在首页、列表页、详细页、其他的业务功能相关页面，都有可能存在同一个功能的入口，如付费咨询、免费咨询业务中，可以直接从首页进入付费咨询入口，也可以通过免费咨询入口再切换到付费咨询入口。每一个入口路径都需要覆盖检查  
5）交互性检查  
一般而言，产生数据和功能交互变化的情况主要有这几个分类：前台<-->前台、后台<-->后台、前台<-->后台。前台从A1页面提交的数据，可能需要在前台A2页面查看到，也会在对应后台的B页面查到记录；后台B1页面修改or添加的数据，对应到前台的A页面产生交互变化，后台本身的不同页面之间也可能存在同一个数据的输出值

**3.版本配置测试**

有时候小程序一次性做了几套不相同的模板，在前端程序代码中修改配置参数，保存后重新编译，即可从一个版本切换到另一版本，同时也需要在管理后台作相应的切换，以保证前端进行数据调用  
对于非公用的部分：不同版本直接的切换，需要保证彼此的功能模块和数据独立性不受干扰影响，即不同版本的管理后台所添加的数据只应该调用到各自对应模板的前台小程序中，不同版本的小程序从前台提交的数据也只会提交到各自管理后台，不应该有交差重叠  
对于公用的部分：切换不同的模板，都会显示相同的内容

**4.兼容性测试**

1）手机操作系统  
常规的手机端OS为：Android（7.x/6.x/4.x/2.x...）、IOS（11.x/10.x/9.x...）  
2）微信版本  
对于已上线的小程序，有可能会因为微信版本升级之后导致对部分小程序的组件支持产生冲突，手机端微信上查看的小程序页面出现样式有异常，比如出现少部分区域的黑屏，这种情况需要同步在小程序的程序包中修改一些组件再次更新

**5.易用性测试**

1）导航  
定位到页面某个模块所在位置，回到顶部or底部，导航条的收展，导航标签的文字是否容易理解  
2）功能入口  
重要且常用业务的功能入口，是否在比较显眼的位置，业务操作过程是否便于大多数用户使用和查看  
3）上下层级进入&返回  
首页<-->列表页、列表页<-->详细页 、首页<-->详细页。不同层级之间的进入和返回实现是否有相应按键易操作  
4）字体、图片、动态交互效果  
字体：标签、标题、内容、动态播放字体...  
图片：轮播图、背景图、封面图、触屏产生的交互图...

#### 5.2 测试准备

#### **1.前端页面**

微信Web开发者工具安装、授权测试用的微信号可预览和调试小程序...  
可参考此文： [微信Web开发者工具-下载、安装和使用图解](https://www.cnblogs.com/zhuochong/p/10436647.html" \t "https://www.cnblogs.com/zhuochong/p/_blank)

#### **2.管理后台**

配置内网测试服务器环境，通过PC端Web站点管理小程序前端的输出内容，可从开发人员获取管理账号进行测试

## 5.3单元测试



|  |
| --- |
| 测试模块：书籍发布 |
| 标识符：1 |
| 测试项：是否能正常跳出页面，是否能上传相关信息 |
| 测试环境：微信app |
| 前置条件：小程序相关数据库等其他运行环境正常 |
| 操作步骤：   1. 点击发布按钮 2. 选择“发布藏书” 3. 弹出发布须知，点击“同意” 4. 录入书籍信息 5. 完成发布 |
| 测试结果：没有出现异常。 |

|  |
| --- |
| 测试模块：书籍预览 |
| 测试项：书籍详情、评论 |
| 测试环境：微信app |
| 前置条件：小程序相关数据库等其他运行环境正常 |
| 操作步骤：   1. 点击首页任意一本书籍 2. 进入详情页面 3. 查看书本价格、发布人等相关信息 4. 添加评论 5. 返回首页 |
| 测试结果：暂时未能实现评论功能 |

## 5.4 集成测试

1.先对主控制模块进行测试，将存根程序代替所有直接附属与主控制模块的模块。

2.每次用一个实际模块代换一个存根程序。

3.在结合进一个模块的同时进行测试

4.在不断进行回归测试

5.重复以上步骤。

未发现各模块借口间存在的问题

## 5.5确认测试

## Alpha测试

### Beta测试

## 5.6白盒测试

暂未进行