|  |  |
| --- | --- |
| 学号 |  |

信 息 工 程 学 院

课程名称: 微信公众平台应用开发实践

|  |  |
| --- | --- |
| 题 目： | 陶票票小程序 |
| 专业班级： |  |
| 姓 名： |  |
| 授课教师： |  |
| 成 绩： |  |

2022 年 5 月 20 日

微信公众平台应用开发实践考核及评分要求

**一、考核要求**

根据给定的选题，设计和实现一个微信小程序

具体要求如下：

1. 微信小程序系统至少包含3个页面，页面跳转逻辑顺畅。
2. 前端技术要求：wxml，wxss，js，微信小程序框架，小程序云开发等。
3. 将开发的过程写成报告，要求内容完整，格式规范，条理清晰。

**二、评分要求**

成绩评定：

平时成绩40%+考核成绩60%

作品具体评分标准如下：

1．选题难易度 10分

2．独立性与创新性 10分

3．程序正确，至少3个页面，样式还原度高、逻辑完整 30分

4. 答辩问题回答正确 20分

5．报告结构清晰、格式符合规范、重点突出、语句通顺 30分

根据作业难度、设计中独立性、创新性、报告书写规范化、作业运行结果及答辩情况综合评分。

**目 录**

[1 设计题目 4](#_Toc5848)

[2 开发环境及开发技术 4](#_Toc25036)

[3 需求分析 4](#_Toc12713)

[4 系统总体设计 5](#_Toc28182)

[4.1 系统的总体功能结构描述 5](#_Toc14903)

[4.2 影片页面详情的功能描述 5](#_Toc331)

[4.3 发现详情的功能描述 6](#_Toc23192)

[4.4 购买详情界面的功能描述 6](#_Toc28740)

[5 详细设计 7](#_Toc759)

[5.1 首页的详细设计 7](#_Toc26221)

[5.2 电影详情页面的详细设计 10](#_Toc25936)

[5.3 购买详情页面的详细设计 14](#_Toc14599)

[5.4 电影页面的详细设计 14](#_Toc10289)

[5.5 发现页面的详细设计 15](#_Toc18803)

[5.6 发现详情页面的详细设计 16](#_Toc20571)

[5.7 数据库的详细设计 17](#_Toc5837)

[6 测试运行 18](#_Toc8300)

[6.1 首页的运行展示 18](#_Toc24926)

[6.2 电影详情页的运行展示 18](#_Toc26377)

[6.3 购买页面的运行展示 19](#_Toc5812)

[6.4 电影页面的运行展示 20](#_Toc14108)

[6.5 发现页面和发现详情页面运行展示 21](#_Toc13151)

[7设计体会 23](#_Toc8982)

[参 考 文 献 24](#_Toc28575)

[附 录 25](#_Toc7016)

[大作业成绩评定表、 30](#_Toc30351)

# 1 设计题目

淘票票小程序

# 2 开发环境及开发技术

硬件环境:处理器 Intel(R) Core(TM) i5-5350U CPU @ 1.80GHz 1.80 GHz 机带 RAM 8.00 GB 系统类型64 位操作系统, 基于 x64 的处理器.服务器使用的是阿里云服务器

开发技术:前端使用html,css,javascipt,微信小程序官方api,后端接口使用javaweb,数据库使用的mysql5.5,容器使用的tomcat8.5。本项目主要以微信小程序界面的书写,js逻辑部分为主。

# 3 需求分析

首先在首页右顶部显示正热播和即将播放按钮,左边有一个按钮点击之后可以选择不同的城市。默认选择的是正在热播,在该界面会展示最近几天正在播放的电影,并且在列表中显示关键的信息。列表分成三部分,左侧显示电影的图片,中间显示电影的标题,电影的标题,电影的导演,主演,根据综合情况,显示今日最热和一周最热,列表的右侧显示购票按钮,点金之后进入到购票页面。

点击即将播放按钮之后出现后面播放的电影。

点击正在热播按钮之后进入到电影的详情页面,在该界面的顶部播放该电影的广告片段,只要用户不离开该界面就循环播放改片段.在电影的下面显示改点影的一些关键信息,如播放时长,播放的具体时间等我.再该部分的下面可以显示评分的关键信息,如评价人数,评价的等级等。再接下来就是显示三个按钮,简介,影评以及更多按钮,在简介部分有电影的详细信息,演职人员列表,演职列表展示演员的图片,以及剧组的剧照等。在最底部显示票房的一些关键数据,如今日票房排名,首周票房,累计票房以及票房详情.在影评部分显示观众热评,在更多部分存放的官方二维码。在每个界面向下划一定的距离之后都会出现回到顶部按钮。

在影院界面显示一些影院的信息,在顶部有选择城市,筛选,综合排序,以及搜索按钮,可以对于价格,距离进行排序。

在发现界面可以看到一些关于电影的界面,这个关于用于的情况进行推荐,点击每一项之后展示影评的详细信息,并且在底部会展示评价内容.在顶部有发送评论的功能,用户点击之后跳转到评论界面,发送评之后跳转回来,并且发送的评论的评论将展示在最前面,对于每一条评论的信息都会展示发布的信息。

# 4 系统总体设计

## 4.1 系统的总体功能结构描述

如下图4-1为陶票票小程序的总体思维导图,总共有八个界面,分别为首页,影院,发现,我的,点击正在热映电影列表部分之后进行到电影详情界面,点击列表中的购买按钮之后进入到购买详情界面,点击发现详情界面之后进入到发现详情界面。点击详情界面底部的发送消息按钮之后进入到发送消息界面。下面的思维导图展示页面之间的相互关系。

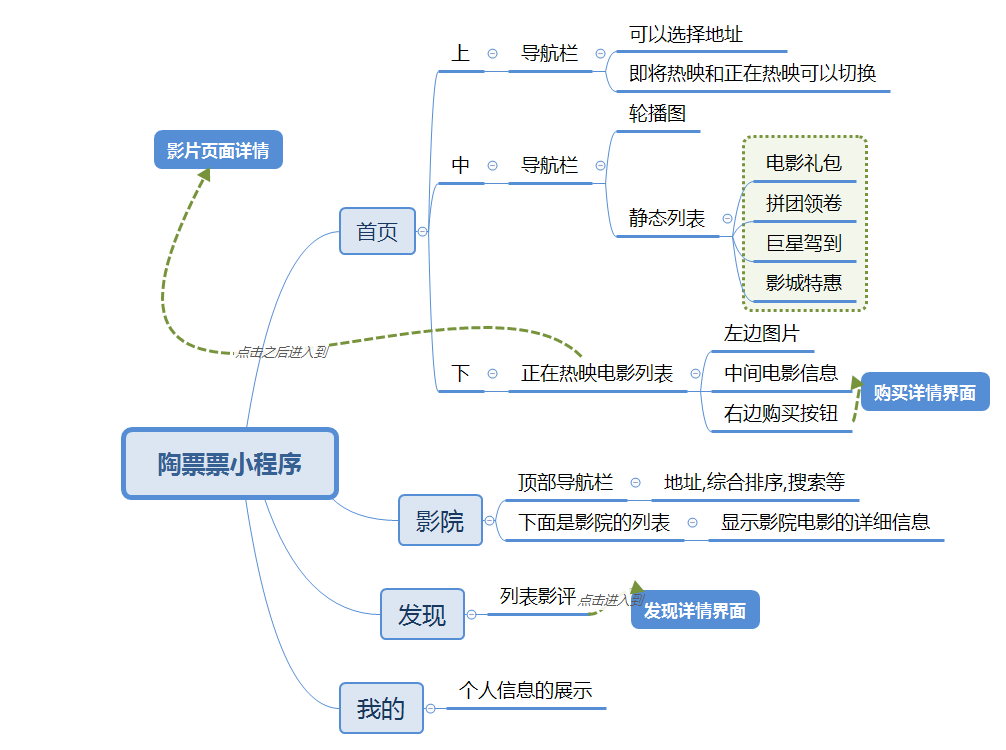


图4-1页面总体结构图

## 4.2 影片页面详情的功能描述

点击首页的电影列表部分之后进入到详情界面,详情界面的功能如下图4-2所示

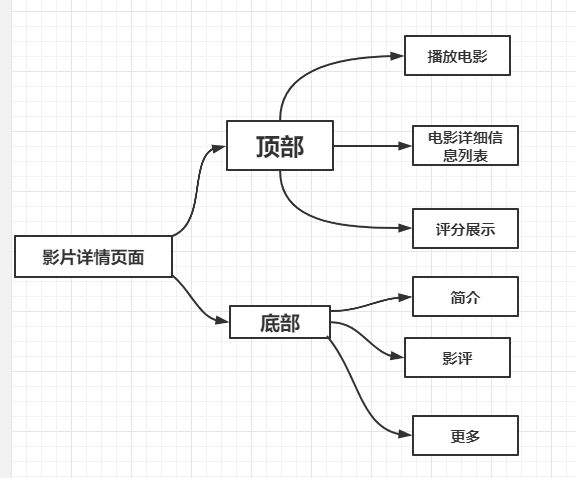


图4-2,影片页面详情

## 4.3 发现详情的功能描述

点击发现界面列表之后进入到发现详情界面,发现详情界的功能面如下图4-3所示。

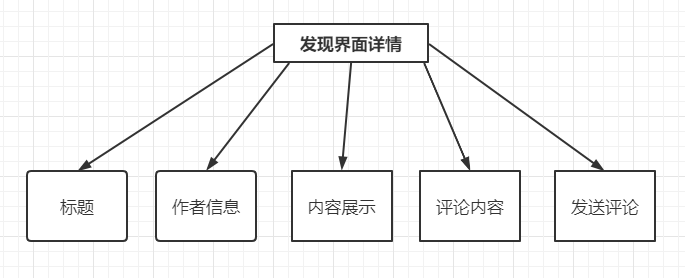


图4-3,发现详情界面

## 4.4 购买详情界面的功能描述

点击首页中的购买按钮之后进入到购买你详情界面,如图4-4所示。

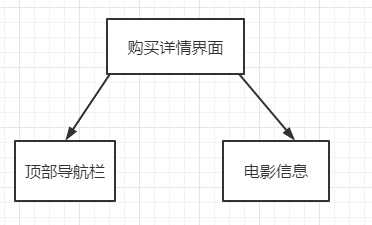


图4-4,购买详情界面

# 5 详细设计

## 5.1 首页的详细设计

首页的样式的布局如图5-2。

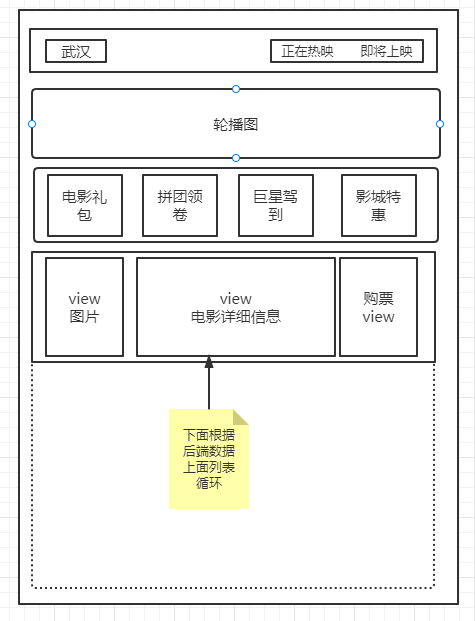


图5-2,首页布局

如上图页面的布局分为四个部分,分别为顶部的导航栏,中间的轮播图,静态列表,以及最下面的动态列表部分,页面的布局主要用是flex布局,后面的界面大多用的是flex布局,将不再做说明。wxml代码结构如下图5-3所示所示

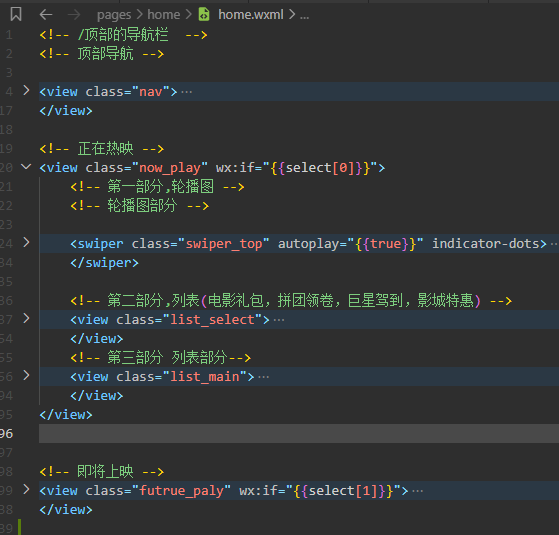


图5-3,首页wxml代码结构

由于样式代码过多,对于一些重复的,常见的将不做详细说明,对于之前不是很熟悉的,单独说明。

为了后期更换主机的时候避免大量改动js中的代码,将主机定义在app.js中,如下所示,在本项目中使用到三个主机地址,baseURL1为本地测试主机地址;baseURL2为本地的最终上线地址,baseURL3为豆瓣官方地址,用于首页和电影详情页面数据的动态渲染。

App({

    globalData: {

        userInfo: null,

        //主机地址

        baseURL1:'http://localhost:8080',

        baseURL2:'http://www.wjiangquan.top',

        //首页的后端请求主机

        baseURL3:'https://m.douban.com'

      }

})

下面的代码用于获取首页数据的后端接口的代码,首页在页面加载的时候通过wx.request向后端发送请求,请求成功的话,将数据通过setData存放在homeList数组中。

    onLoad(options) {

        this.getRequest();

    },

    // 发送请求

    getRequest(){

        wx.request({

          url: this.data.baseURL3+'/rexxar/api/v2/subject\_collection/movie\_showing/items',

          method:'GET',

          success:res=>{

             this.setData({

                 homeList:res.data.subject\_collection\_items

             })

          }

        })

},

下面列举部分wxml代码,通过下面的方式对于数据进行动态渲染。

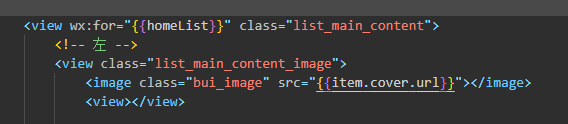


图5-4,数据动态渲染

本页面的主要难点部分在于在于正在热映和即将上映之间的切换,下面将详细说明该功能的实现代码。下面给出关键的wxml的代码以及js代码,为了避免篇幅过长就不列出wxss的代码,样式可以根据自己的喜好进行设置,在后文也尽可能少的列出wxss代码。wx:for对于js中data中的数据进行渲染,wxml中将出现并列的即将上映和正在上映view标签,同时将在页面中进行渲染数据。在js中data初始数组中的select中默认为true和false,这个时候展示的是正在热映的内容。当点击正在热映或者即将上映按钮的时候将触发navClick函数,这个时候通过e.currentTarget.dataset.index获取在自定义的属性data-index中的数据index(index的值由于wx:for的存在将默认从0开始增加)来确定点击的是正在上映还是即将上映,然后改变select中的true和false的顺序,然后通过select[index]?’select’:’’给点击的那一个设置样式。并且通过wx:if来隐藏未被点击的,显示点击的那一项。

wxml的代码如下图5-5所示。

<view class="nav\_right">

        <view class="nav\_list {{select[index] ? 'select' : ''}}" wx:for="{{navList}}" wx:key="index" bindtap="navClick" data-index="{{index}}">{{item}}</view>

 </view>

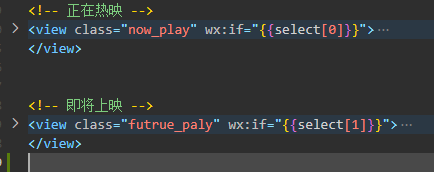


图5-5,正在热映和即将上映的代码

js中的关键代码如下

  data: {

        baseURL3:app.globalData.baseURL3,

        navList:["正在热映","即将上映"],

        select:[true,false],

    },

 // 切换导航栏时(获取点击时自定义的样式)

    navClick(e){

        let index = e.currentTarget.dataset.index;

        if(index === 0){

            this.setData({

                select:[true,false]

            })

        }else{

            this.setData({

                select:[false,true]

            })

        }

        console.log(index);

    },

5.2 电影详情页面的详细设计

当点击首页中的电影列表部分的时候,将跳转到电影详情页面。电影详情页面的样式布局如图5-6所示。

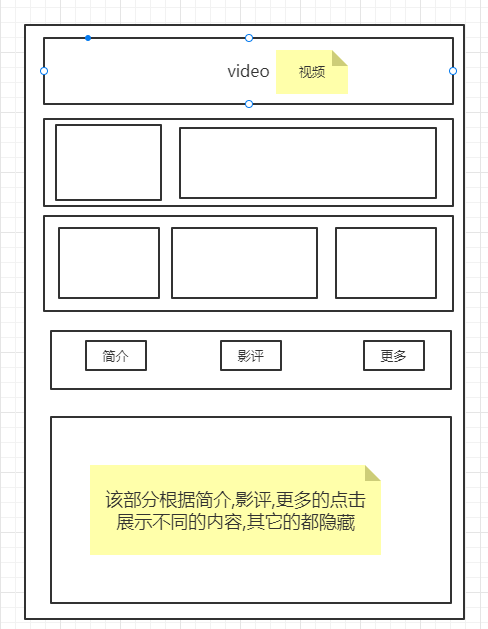


图5-6,电影详情页面设计图

如上图所示,该界面大体上主要分为5个部分,第一部分展示宣传视频部分,第二部分展示该电影的列表信息,第三部分展示电影的评分信息,第四部分展示导航列表,最后一部分展示导航列表对应的内容部分。

本节讲解的的第一个点为从首页点击之后进入到详情页面,如何保证进入到所点击的列表项所对应的详情内容。

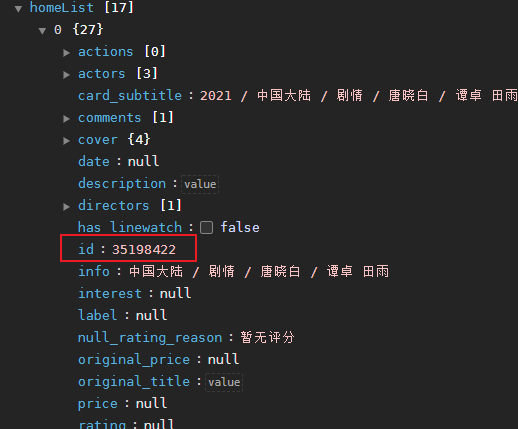
如下为首页关键的wxml代码,为了避免内容过于冗余,这里仅列出了关键的代码,当点击某一项的时候,通过自定义属性data-id获取所点击的列表项的后端传过来的id如图5-9所示。并将id传到电影详情页面如图5-10所示。如何5-11,电影详情页面通过传过来的id向后端发送请求,获取对应页面的数据。如图5-11是电影详情页面的部分接口数据。



图5-7首页渲染的数组



图5-8 列表项的中间部分



5-9首页的后端接口

 // 跳转到电影的详情页面

    toDetail(e){

        let id = e.currentTarget.dataset.id;

        wx.navigateTo({

          url: '/pages/movieDetail/movieDetail?id='+id,

        })

    },

5-10 首页toDetail函数

  // 页面加载

    onLoad: function (options) {

        let id = options.id;

        console.log("传过来的id",id);

        let labelUrl = "https://m.douban.com/rexxar/api/v2/movie/"+id+"/tags?count=8";

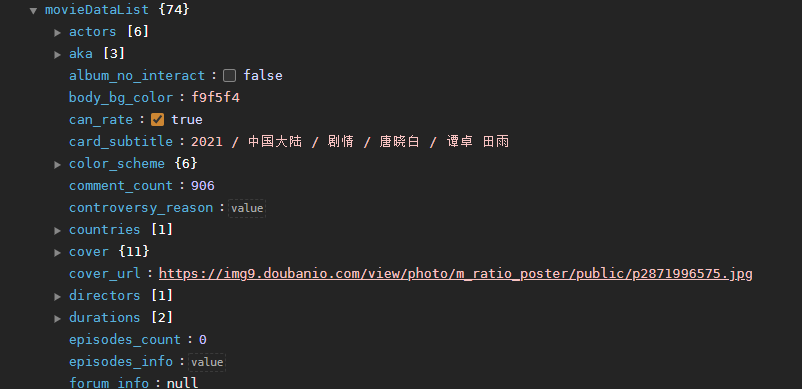
        this.getMovieData(id);

        this.getCommentData(id);

        this.getMovieTagData(id);

    },

5-11电影详情页面接收首页得到传过来的id



5-12 电影详情页面的部分接口数据

如上图5-11所示,在电影详情页面中的下方点击简介,影评所对应的请求函数。

回到顶部的代码如下所示。如图5-13为wxml的代码。

<image src="/images/temporary/top.png" class="go\_top" hidden="{{!go\_top\_show}}" bindtap="goTop"></image>

图5-13回到顶部wxml代码

  // 获取滚动条当前位置

     onPageScroll: function (e) {

        // console.log(e)

        if (e.scrollTop > 300) {

            this.setData({

                go\_top\_show: true

            })

        } else {

            this.setData({

                go\_top\_show: false

            })

        }

    },

    // 回到顶部

    goTop: function (e) {

        if (wx.pageScrollTo) {

            wx.pageScrollTo({

                scrollTop: 0

            })

        } else {

            wx.showModal({

                title: '提示',

            })

        }

    },

5.3 购买详情页面的详细设计

电影详情页面的样式布局如图5-6所示。

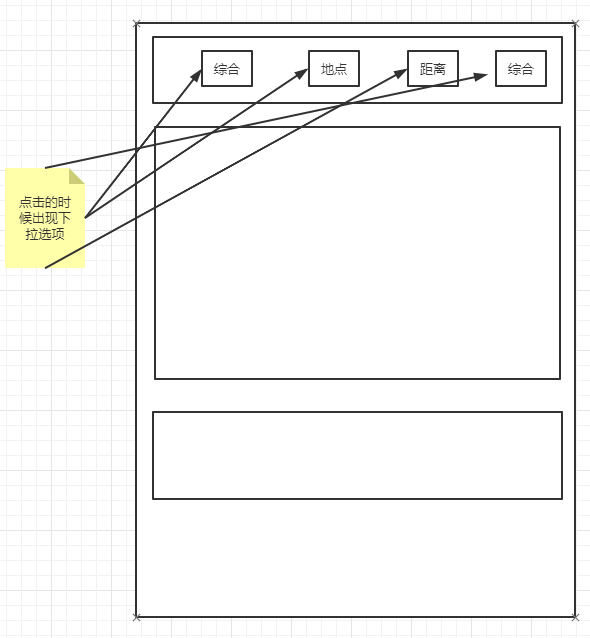


图5-6,电影详情页设计图

对于本页面主要主要不同前面的地方在于导航栏部分的动画,由于动画是用css写的,代码过多,这里就不展开了。

5.4 电影页面的详细设计

如下图5-7,为电影页面的详细设计。

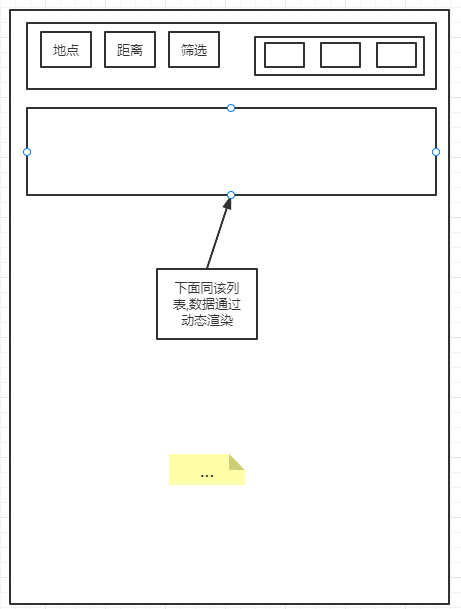
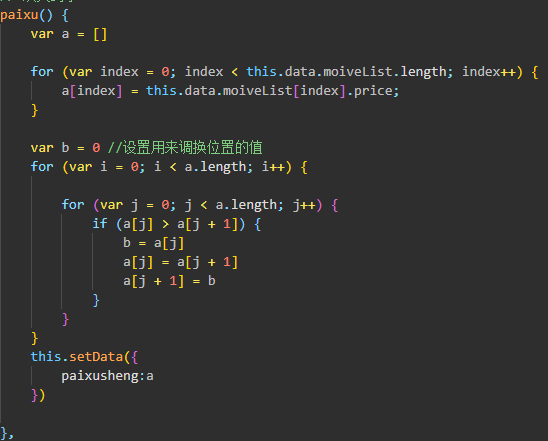


图5-7,电影页面详细设计

在该界面中主要的技术点在对于从后端获取的数据通过点击不同得到按钮,对价格进行升序或降序的排列,让用户有更好的体验。主要wxml和js的代码如下所示。

 <text bindtap="paixu" class="iconfont icon-sanjiaoxing\_shang"></text>

 <text bindtap="paixu1" class="iconfont icon-sanjiaoxiangshangjiantou"></text>

5.5 发现页面的详细设计

如下图5-8为发现页面的详细设计,左边为图片右边为标题和作者用户名的展示。

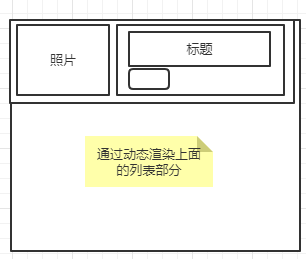


图5-8,发现页面详细设计

发现页面主要的js代码如图5-9所示

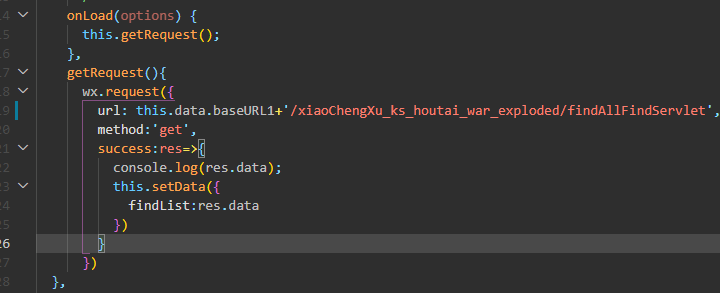


图5-9,发现界面发送请求的代码

对于标题部分当超过两行的时候多余的内容显示成三个点的css代码如下图5-10所示。首先给两行文字设置整体的高为100px,然后行高为高的一半,当再来的一行就会超过100px,就会显示成三个点。

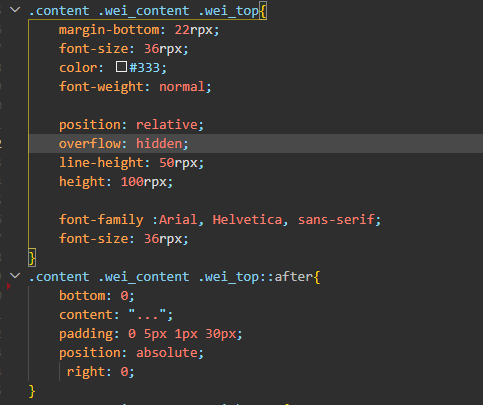
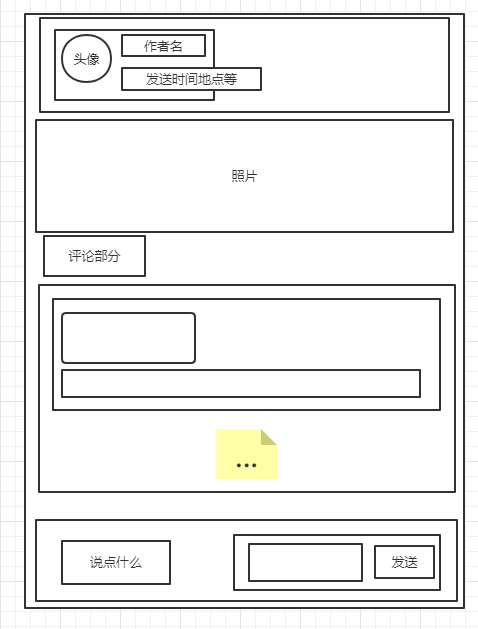


图5-10,多余内容显示点的wxss代码

5.6 发现详情页面的详细设计

如图5-10,为发现详情界面的详细设计图,其中顶部的发现页面点击的那一列所传过来的头像和作者的名字,照片也是发现页面传过来的。



5-10,发现详情页面设计图

关键的逻辑代码分析,如下图5-11所示

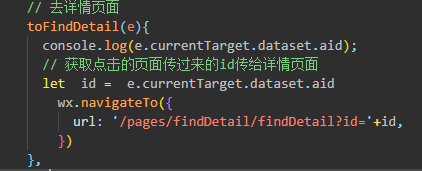
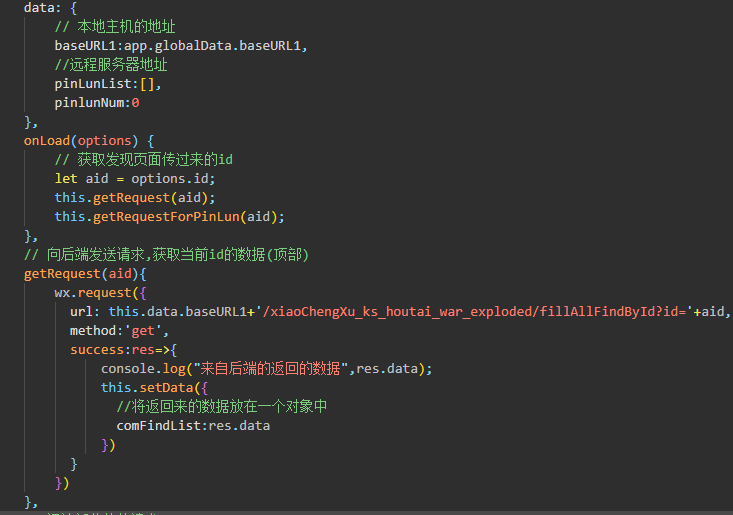
 

图5-11,发现详情页面的关键js代码

如下图5-12为获取评论的内容,并将评论的内容放在pinLunList这个数组中,getCommentNumber可以计算评论的数量。

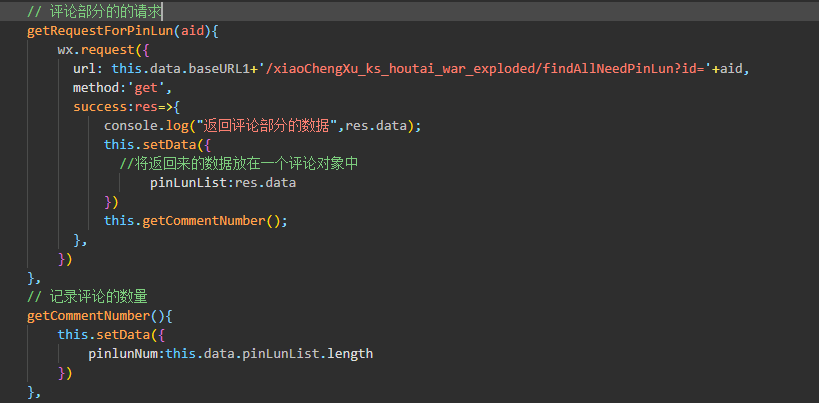


图5-11,获取评论内容,并统计评论的数量

5.7 数据库的详细设计

下面列出了主要的二张表的设计,如图5-12为find页面的表结设计,如图5-13为发现详情页面评论部分的表结构。图5-14为数据库中的表。

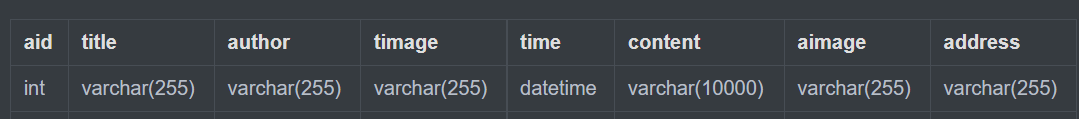


图5-12,发现界面表结构

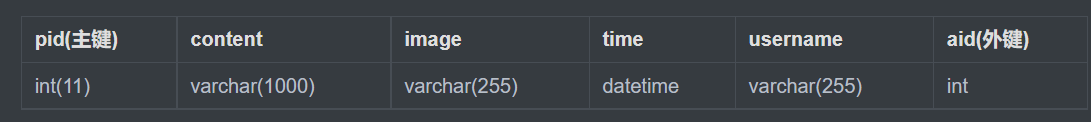
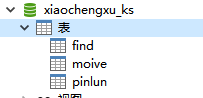


图5-13,评论部分的表结构



# 6 测试运行

## 6.1 首页的运行展示

如图6-1为进入小程序的时候展示效果图6-2当点击即将上映时的展示效果,顶部的轮播图每隔几秒会刷新一次,当往下滑动的时候会刷新出更多的列表,首页的正在热映和即将上映和豆瓣的实时同步。

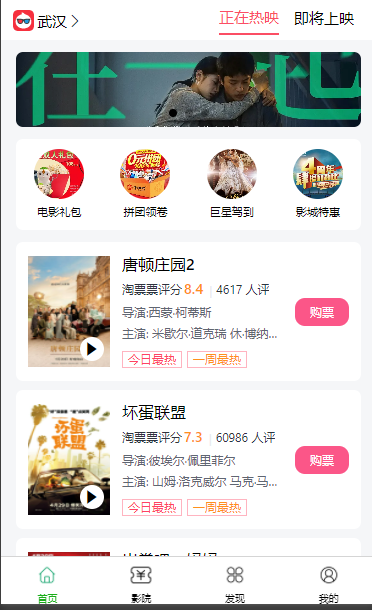
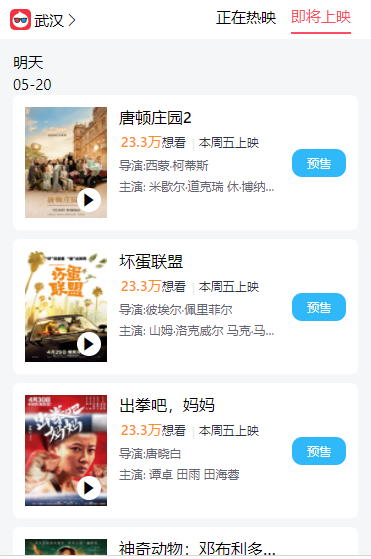
 

图6-1,正在热映运行效果 图6-2,即将上映的运行效果

## 6.2 电影详情页的运行展示

如下图中的6-3为电影详情页的顶部运行效果,上面为电影宣传广告部分,6-4和6-5为电影简介部分运行效果,点击顶部可以回到顶部,演职人员可以向右滑动展现所有的演职人员。

图6-3,详情页顶部 图6-4,详情页简介 图6-5,详情页简介

如图6-6为影评部分运行效果,6-7为更多部分运行效果

图6-6,影评部分效果 6-7,更多部分效果

## 6.3 购买页面的运行展示

当点击购买按钮的时候将跳转到购买页面。如下图中6-8所示。

图6-8,购买界面的运行效果

点击不同的导航栏的时候的运行效果如下图6-9,6-10,6-11和6-12所示。

图6-9,点击评价的效果展示 图6-10,地点的效果展示

6-11距离的效果展示 6-12,综合的效果展示

## 6.4 电影页面的运行展示

如下图7-13和7-14为影院界面当点击筛选右边的降序按钮时的界面的运行效果。图7-9为影院界面当点击筛选右边的升序序按钮时的界面的运行效果。

图7-13,影院价格降序运行效果 图7-14,价格升序运行效果

## 6.5 发现页面和发现详情页面运行展示

如图7-15为发现界面的运行效果,图7-16为发现详情界面的运行效果。

图7-15,发现界面运行效果 图7-16,发现详情页面

下图7-17为评论部分的运行效果。当点击发送之后会条装到图7-18的发送评论部分。发送的内容将在评论的最前面进行显示。

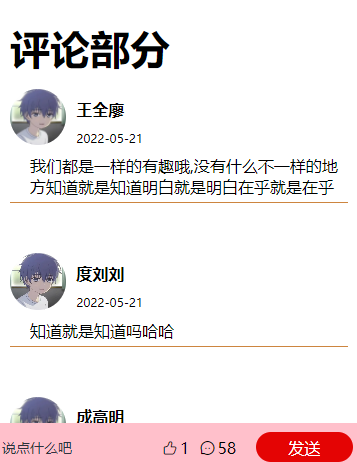
 

图7-17,评论部分的运行效果 图7-18 发送评论部分运行效果

如图7-19当输入内容之后的效果,7-20发送评论之后默认展示在第一条的运行效果。

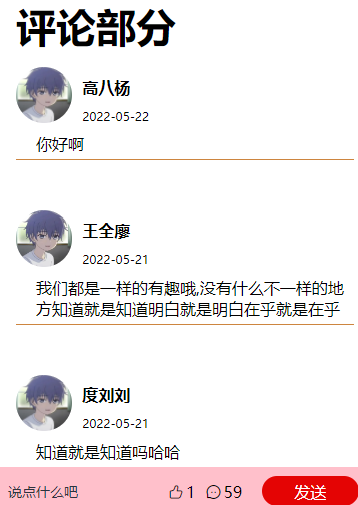
 

图7-19,输入评论内容 7-20,发送评论之后的效果展示

# 

# 7设计体会

对于本次课程设计,从前期的准备到最终选定题目,到需求的确定,直到最终将产品实现,花了差不多4个晚上的时间,自己独自从零到一完成了整个项目的开发,从需求的确定,到从页面的设计,到页面布局的设计,到数据库的设计,以及后端接口的编写,整个过程中充满了各种挑战,但是最终的效果确实让自己满意的,比较有成就感的过程,在本次课程设计的过程中,也接触到了之前没有接触到的技术,写了一些之前没有写过的代码,发现了一些好的样式书写的技巧。同时也遇到的一些困难,但是更多的是进步与新知识的积累。

先从本课程主要部分前端页面样式和js逻辑部分开始说起吧,在编写微信小程序前端代码的过程中比较繁琐的地方就是给页面的文字和背景进行取色,由于自己没有专门的UI设计师提前做好页面的静态效果图,所以自己就通过陶票票的的手机网页面端对着界面进行逐个取色,这个过程花的时间比较长。也是从这一刻开始意识到uI设计师的重要性,它能提高前端代码的书写效率,同时也可以让做出来的页面的审美更高。也就是在这个过程中接触到之前没有接触到的一些页面效果,但是又比较常见的,比如当文字过多的时候多余的文字用三个点进行显示,在首页列表和发现界面列表展示的时候有用到,并且还可以超过固定的行数的时候,多余的显示成三个点。还另一个点就是多余的的文字通过展开和收起按钮进行控制文字的,在电影详情页面有用到,对于这个这个效果自己在刷手机的过程中也经常遇到过,通过本次效果让自己明白亲自探究了它的实现原理。

在本次小程序项目中,虽然是前端项目,但是由于找到的陶票票的接口不是很完整,只有首页和电影详情页面的接口。所以自己就设计了三张表,通过javaweb编写了其它几个页面的接口。最终也达到了自己比较满意的结果,在整个过程中还算比较顺利。

对于本次项目,由于时间的限制,项目还有许多不足的地方,比如发布文章功能,购票功能等。希望后面可以将它们逐步完善。

# 参 考 文 献

[1] https://www.runoob.com/

[2] <https://mp.weixin.qq.com/>

[3] 佚名. 微信小程序[J]. 上海信息化, 2017(01):9-10.

[4]刘红卫. "微信小程序应用探析." 无线互联科技 23(2016):3.

[5]林晓艳, 王军峰, 王文军,等. 微信小程序的用户体验研究[J]. 工业设计研究, 2017(1):6

附 录

由于代码过多,该部分只是展示部分代码

完整代码地址: https://github.com/JordanFoever/xiaoChengXu\_ks

home.js代码如下

const app = getApp();

Page({

data: {

baseURL3:app.globalData.baseURL3,

navList:["正在热映","即将上映"],

select:[true,false],

imageList:[

"../../image/1.jpg",

"../../image/2.jpg",

"../../image/3.jpg",

"../../image/4.jpg"

]

},

onLoad(options) {

this.getRequest();

},

// 发送请求

getRequest(){

wx.request({

url: this.data.baseURL3+'/rexxar/api/v2/subject\_collection/movie\_showing/items',

method:'GET',

success:res=>{

this.setData({

homeList:res.data.subject\_collection\_items

})

}

})

},

// 跳转到电影的详情页面

toDetail(e){

let id = e.currentTarget.dataset.id;

wx.navigateTo({

url: '/pages/movieDetail/movieDetail?id='+id,

})

},

// 切换导航栏时(获取点击时自定义的样式)

navClick(e){

let index = e.currentTarget.dataset.index;

if(index === 0){

this.setData({

select:[true,false]

})

}else{

this.setData({

select:[false,true]

})

}

console.log(index);

},

headleBuy(){

wx.showModal({

title:"确定要买票吗",

// content:"还有余票",

cancelColor: 'blue',

cancelText:"我再想想",

confirmText:"去购买页",

// editable:true,

// placeholderText:"请输入年龄",

success:res=>{

if(res.confirm){

wx.navigateTo({

url: '/pages/buy/buy',

})

}

}

})

},

headleYuShou(){

wx.showModal({

title:"预售",

// content:"还有余票",

cancelColor: "red",

cancelText:"我再想想",

confirmText:"去预售页面",

// editable:true,

// placeholderText:"请输入年龄",

success:res=>{

if(res.confirm){

wx.navigateTo({

url: '/pages/buy/buy',

})

}

}

})

}

})

|  |  |  |  |
| --- | --- | --- | --- |
| 大作业题目 | **淘票票小程序** | | |
| 学生答辩或质疑记录:  问题1:在电影详情页面中让评论数在影评右上角是怎么做到的?  答:通过定位,影评放在一个view标签中,数字放在一个text标签中,text标签是view标签的子元素,view使用相对定位,text使用绝对定位,然后通过top和right属性将数字设置到右上角。  问题2:在本次项目中，使用最多的布局方式是什么。为什么使用它?  答: 使用最多的布局方式为flex布局，也就是弹性布局，对于这种布局方式，可以应对不同屏幕宽度不同设备的一整套新的布局方案。主要是对一个容器中的子元素进行排列、对齐和分配空白空间的方案的调整。并且我将其封装在一个css文件中，这样项目的结构更加的清晰，使用起来更加的简单。  问题3:在wx.request()方法中当请求成功要使用this.setdata来保存数据?  答:因为如果直接使用data对象的话，只是拿到data对象中的数据，而在请求成功后，需要将请求成功后的数据放入到data对象中,我们希望改变data中的数据，就需要通过this.setdata的方式将拿到的数据赋给data对象中的变量。 | | | |
| **评 分 依 据** | | **分 值** | **评分成绩** |
| 1．选题难易度 | | 10 分 |  |
| 2．独立性与创新性 | | 10 分 |  |
| 3．程序正确，至少3个页面，样式还原度高、逻辑完整 | | 30分 |  |
| 4. 答辩问题回答正确 | | 20分 |  |
| 5．报告结构清晰、格式符合规范、重点突出、语句通顺 | | 30分 |  |
| 总 分 | | 100分 |  |
| 指导老师签字：  年 月 日 | | | |

**大作业成绩评定表**、