**跨端框架应用编程**

**结课考试**

|  |  |
| --- | --- |
| **题 目** | **美发预约APP** |
| **学 号** | **1913040716** |
| **姓 名** | **徐昕沛** |
| **联系方式** | **18734660966** |
| **提交日期** | **2021.12.27** |

目 录

[第1章 需求分析 4](#_Toc91506406)

[1.1 功能概述 4](#_Toc91506407)

[1.1.1登录模块 4](#_Toc91506408)

[1.1.2首页模块 4](#_Toc91506409)

[1.1.3预约模块 5](#_Toc91506410)

[1.1.4购物车模块 5](#_Toc91506411)

[1.1.5我的模块 5](#_Toc91506412)

[1.2 需求模型 6](#_Toc91506413)

[1.2.1登录模块 6](#_Toc91506414)

[1.2.2首页模块 7](#_Toc91506415)

[1.2.3预约模块 7](#_Toc91506416)

[1.2.4购物车模块 8](#_Toc91506417)

[1.2.5我的模块 10](#_Toc91506418)

[第二章 原型设计 11](#_Toc91506419)

[2.1登录模块 11](#_Toc91506420)

[2.2 首页模块 12](#_Toc91506421)

[2.3预约模块 13](#_Toc91506422)

[2.4购物车模块 14](#_Toc91506423)

[2.5我的模块 15](#_Toc91506424)

[第三章 APP实现 19](#_Toc91506425)

[3.1 项目框架 19](#_Toc91506426)

[3.2 功能实现 20](#_Toc91506427)

[3.2.1登录模块 20](#_Toc91506428)

[3.2.2首页模块 24](#_Toc91506429)

[3.2.3预约模块 34](#_Toc91506430)

[3.2.4购物车模块 42](#_Toc91506431)

[3.2.5我的模块 48](#_Toc91506432)

[第四章 APP打包 75](#_Toc91506433)

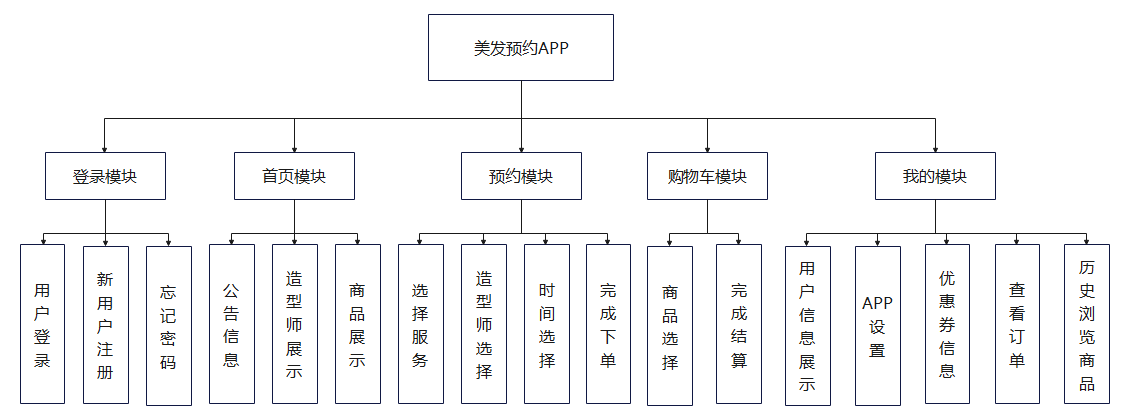
[4.1 打包过程 75](#_Toc91506434)

[4.1.1打包为原生App（云端） 75](#_Toc91506435)

[4.2设计资源 77](#_Toc91506436)

# 第1章 需求分析

## 1.1 功能概述



**1.1功能模块图**

### 1.1.1登录模块

本模块主要是要实现美发预约APP的登录功能，用户点击进入APP以后需要先进行登录才可以进行接下来一系列的操作。在该登录页面中，用户可以通过在输入框中输入自己的手机号和密码，系统经过验证后成功登陆进入APP页面中。模块会判断用户输入的手机号否符合格式，如不符合格式会弹出提示框提醒用户重新输入正确格式的手机号。同时，如果是新用户，也可以通过点击注册按钮点击跳转进入注册界面进行注册。老用户如在登陆的过程中忘记了自己的密码，也可以点击忘记密码选项进入忘记密码界面进行密码的找回或者重设密码。本系统同时第三方账号登陆，如果用户之前手机号已绑定微信或者qq，则可以通过页面底部的提示按钮选择相应的方式登录，大大提高了用户使用系统的便捷性，方便用户更加快捷简洁的进入系统。

### 1.1.2首页模块

本模块主要是要实现美发预约APP的首页展示功能，用户进入APP后首先看到的就是首页这一主页面。从上往下来说，最上方的导航栏可以通过选择店铺进入到不同的店铺界面，并配有搜索功能进行搜索用户想要的服务、造型师或商品。下方的轮播图主要是对美发店的店面进行图片的展示。公告栏区域则是通过滚动的形式，将美发店的实时咨询通告给用户。下方则是提供了一个快速跳转功能，用户可以通过点击图标快速跳转至相应的页面，如在“关于我们”这一页面中，用户就可以看到该APP服务的美发店的位置与简介。其他页面分别可以跳转至预约界面和订单界面。本页面最后的内容则是对明星造型师和相关商品的展示，用户可以了解到造型师的信息，并且也能了解到美发店中所出售的商品信息，如价格及付款人数。

### 1.1.3预约模块

本模块主要是要实现美发预约APP的预约功能，在本界面中，用户可以通过选择相应的信息完成美发订单的预约。用户首先可以看到自己的信息展示，系统会显示出用户的账号余额和相关的个人信息。用户可以通过点击图标选择自己要预约的服务，如“染发”、“剪发”、“烫发”、“洗发”等，并通过点击“更多服务>”来查看选择其他服务。下方则是造型师的选择，用户通过多列选择器完成对预约的店铺和造型师进行选择，不同的店铺会有不同的造型师，并且界面会将选择后的结果进行展示。用户最后需要选择预约的时间，点击相应的选项之后，用户需要在时间选择器中选择日期和时间。最后用户点击“立即下单”按钮完成订单的预约，并跳转至预约成功界面。在预约成功界面中，用户可以点击左上角的返回按钮返回预约界面，也可以点击界面提示的按钮返回至首页，完整的实现了该模块的功能。

### 1.1.4购物车模块

本模块主要是要实现美发预约APP的购物车功能，主要是对用户选择进购物车的商品进行展示。用户可以在这一模块之中看到已选择商品的相关信息，如商品的图片、品名和价格。用户可以对该界面中的商品进行一系列操作，如用户可以在左侧的选项栏中选择所需要购买的商品，同时也可以选择需要购买的件数。若用户不需要购买此商品，则可以通过删除按钮将该商品从购物车中删除。商品下方提供了一个“批量操作”按钮，用户可以点击该按钮完成对商品的批量操作。界面的最下方一栏中提供了全选按钮，若用户购物车中的商品较多，点击该按钮可以实现商品的全选或取消，方便用户操作。用户可以通过是显示用户购买商品的价格，系统会自动计算用户选择商品的总价，最后用户可以通过点击“去结算”按钮跳转至结算界面，付款完成商品的购买。

### 1.1.5我的模块

本模块主要是要实现美发预约APP的个人信息展示的功能，主要是完成用户查看自己的用户信息、对APP的设置、查看订单信息以及历史浏览商品等相关功能。该界面的最上方，是用户个人信息的简单展示，主要有用户名、用户头像和使用天数。同时也展示了会员卡的余额、优惠券以及累计消费。用户可以通过点击右侧“齿轮”按钮，跳转至设置界面。在此界面中用户可以设置收货地址，并且选择是否接受该APP的消息推送，或者点击最下方的按钮“点击退出登录”，跳转返回至登录界面。用户也可以点击优惠券查看自己拥有的优惠券并使用。下方主要是订单页面的展示，用户可以在此看到自己的预约订单和购物订单。通过分段器对用户不同状态的订单进行了区别，方便用户对订单的管理。我的界面的最下方设置了一个用户历史浏览商品的横向滚动列表查看自己曾经浏览过的商品。

## 1.2 需求模型

### 1.2.1登录模块

正常处理

⽤例名称：登录模块

场景 ：

Who：用户

Where：登陆界面

When:App正常运行且用户登陆时

⽤例描述

1. 用户打开App进入登陆界面。

2. 输入手机号和密码。

3. 用户点击登录按钮服务器验证手机号和密码。

4. 服务器验证通过，用户进入系统。

⽤例价值：用户成功登录进入系统首页进行相关操作

约束和限制 ：

合规性：电话号须符合标准格式，密码需要为6-12位字母数字组合

技术性：界面使⽤中⽂，因为用户都是中国⼈。

安全性：手机号和密码必须通过验证。

可靠性：App可以正常运行，⾄少⼀天内不要出现问题。

异常处理

异常步骤

步骤3用户点击登录按钮服务器验证手机号和密码

3.1 手机号未输入，需要输入手机号

3.2 密码未输入，需要输入密码

3.3 密码不符合6-12位字母数字组合

步骤4服务器验证通过，用户进入系统。

4.1 手机号不存在，需要用户注册

4.2 密码错误，需要用户找回密码

异常：指的是流程的异常情况，⽽⾮系统的异常情况。

替代处理

替代步骤

步骤2: 输入手机号和密码。

2-A：用户点击QQ登录，使用QQ登录

2-B：用户点击微信登录，使用微信登录

### 1.2.2首页模块

正常处理

⽤例名称：首页模块

场景 ：

Who：用户

Where：首页界面

When:App正常运行且用户登陆成功

⽤例描述

1. 用户登陆完成后进入首页。

2. 用户可以通过点击相应的提示完成操作。

⽤例价值：用户成功登录进入到系统首页并可以进行相关操作。

约束和限制 ：

合规性：用户只有查看权限不能修改首页中的内容。

技术性：界面使⽤中⽂，因为用户都是中国⼈。

安全性：用户通过验证成功登录进入系统。

可靠性：App可以正常运行，⾄少⼀天内不要出现问题。

异常处理

异常步骤

步骤1用户登陆完成后进入首页

1.1用户登录失败不能登陆进入首页

异常：指的是流程的异常情况，⽽⾮系统的异常情况。

### 1.2.3预约模块

正常处理

⽤例名称：预约模块

场景 ：

Who：用户

Where：预约界面

When:App正常运行且用户登录成功后选择进入预约界面时

⽤例描述

1. 用户成功登录进入系统。

2. 用户在首页中点击跳转进入到预约界面。

3. 用户在界面中选择相应的服务、造型师、时间。

4. 用户点击最下方的立即下单按钮完成下单。

5. 用户跳转至预约成功界面完成订单预约。

⽤例价值：用户成功完成订单的预约。

约束和限制 ：

合规性：用户所选时间需为店铺营业时间。

技术性：界面使⽤中⽂，因为用户都是中国⼈。

安全性：用户通过验证成功登录进入系统。

可靠性：App可以正常运行，⾄少⼀天内不要出现问题。

异常处理

异常步骤

步骤4用户点击最下方的立即下单按钮完成下单。

4.1 服务选项未选择，需要选择服务项。

4.2 造型师未选择，需要选择造型师。

4.3 时间未选择或选择时间不在营业时间内，需要重新选择。

异常：指的是流程的异常情况，⽽⾮系统的异常情况。

替代处理

替代步骤

步骤2: 用户在首页中点击跳转进入到预约界面。

2-A：用户点击首页预约按钮跳转至界面。

2-B：用户点击底部tabbar图标跳转至界面。

### 1.2.4购物车模块

正常处理

⽤例名称：购物车模块

场景 ：

Who：用户

Where：购物车界面

When:App正常运行且用户登录成功后选择进入购物车界面时

⽤例描述

1. 用户成功登录进入系统。

2. 用户在首页中点击跳转进入到购物车界面。

3. 用户在界面中选择相应的商品。

4. 用户选择商品的件数。

5. 用户点击“去结算”按钮跳转至订单结算页面。

⽤例价值：用户成功完成商品的结算。

约束和限制 ：

合规性：用户所选商品数量不可以超过库存。

技术性：界面使⽤中⽂，因为用户都是中国⼈。

安全性：用户通过验证成功登录进入系统。

可靠性：App可以正常运行，⾄少⼀天内不要出现问题。

异常处理

异常步骤

步骤4.用户选择商品的件数。

4.1 用户选择商品件数为负数，系统提示不能再少了

4.2 用户选择商品件数超过库存，无法下单。

步骤5. 用户点击“去结算”按钮跳转至订单结算页面。

5.1 用户未选择任何商品，需要选择商品下单结算。

异常：指的是流程的异常情况，⽽⾮系统的异常情况。

替代处理

替代步骤

步骤3. 用户在界面中选择相应的商品。

3-A：用户点击商品前的按钮选择商品。

3-B：用户点击底部全选按钮全部选择商品。

3-C：用户点击批量操作按钮选择商品。

### 1.2.5我的模块

正常处理

⽤例名称：我的模块

场景 ：

Who：用户

Where：我的界面

When:App正常运行且用户登陆成功

⽤例描述

1. 用户登陆完成后进入首页。
2. 用户点击底部tabbar图标进入我的界面

3. 用户可以通过点击相应的提示完成操作。

⽤例价值：用户成功登录进入到我的界面并可以进行相关操作。

约束和限制 ：

合规性：用户只有查看权限不能修改界面中的内容。

技术性：界面使⽤中⽂，因为用户都是中国⼈。

安全性：用户通过验证成功登录进入系统。

可靠性：App可以正常运行，⾄少⼀天内不要出现问题。

异常处理

异常步骤

步骤1用户登陆完成后进入首页

1.1用户登录失败不能登陆进入首页

异常：指的是流程的异常情况，⽽⾮系统的异常情况。

# 第二章 原型设计

## 2.1登录模块

登陆界面：

****

**2.1登录界面原型设计**

## 2.2 首页模块

首页界面：



**2.2.1首页界面原型设计**

关于我们界面：



**2.2.2关于我们界面原型设计**

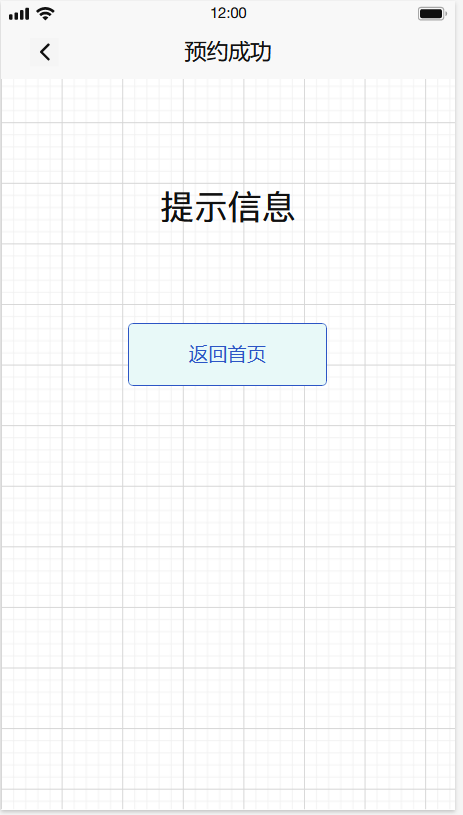
## 2.3预约模块

预约界面：

****

**2.3.1预约界面原型设计**

预约成功界面：

****

**2.3.2预约成功界面原型设计**

## 2.4购物车模块

购物车界面：

****

**2.4.购物车界面原型设计**

## 2.5我的模块

我的界面：



**2.5.1我的界面原型设计**

优惠券界面：



**2.5.2优惠券界面原型设计**

设置界面：



**2.5.3设置界面原型设计**

收货地址界面：



**2.5.4收货地址界面原型设计**

添加用户地址界面：



**2.5.5添加用户地址界面原型设计**

预约订单界面：



**2.5.6预约订单界面原型设计**

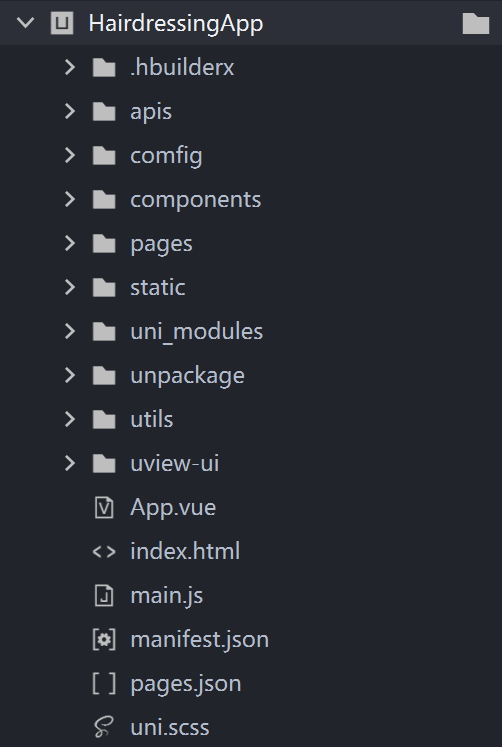
商品订单界面：



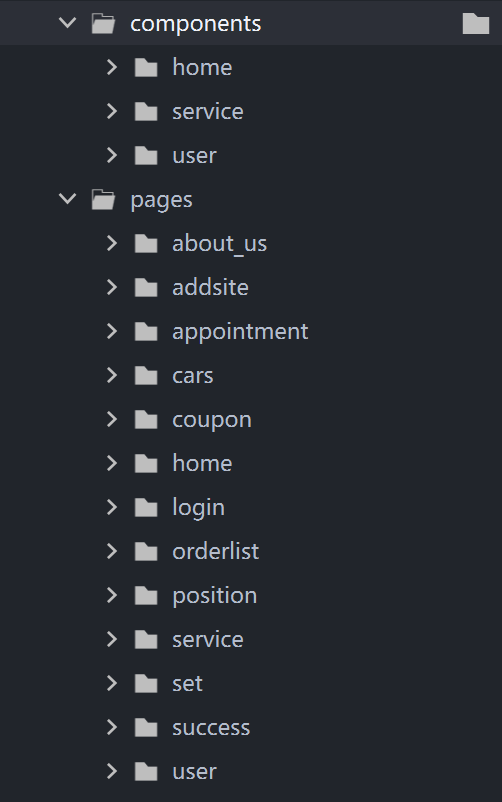
**2.5.7商品订单界面原型设计**

# 第三章 APP实现

## 3.1 项目框架



**3..1.1项目框架**



**3.1.2项目框架**

pages.json ：用来对 uni-app 进行全局配置,设置底部tabbar以及窗口样式等其他功能。

uview-ui：主要是多平台快速开发的UI框架，本次项目大量使用了其中的组件。

static:使用到的静态本地资源，存放要用到的图片。

components:所有用到的组件文件夹。

pages:所有页面存放的文件夹。

APP.vue : 用来声明公共的代码，比如有css样式，是否使用一些其他的组件（路由等）

main.js：用来声明公共的js代码。

## 3.2 功能实现

### 3.2.1登录模块

实现思路：

该模块主要实现程序的登录功能，首先要求设置出两个输入文本框，分别负责用户的手机号和密码输入。同时在这一板块中要求要对手机号和密码进行判断，判断是否输入手机号或密码，手机号格式是否符合标准格式。对于密码的输入同时也需要做隐藏处理，剩下的要求则是需要定义好登录和注册按钮，用户输入正确的手机号和密码后可以跳转进入APP首页，注册按钮跳转至注册，可以通过点击“忘记密码”文字找回密码。下方设置图标提供其他第三方登录方式。最后设置好背景图和相应的样式使界面更加美观。

关键代码:

login.vue

<!-- 登录页 -->

<template>

<view style="height:100vh; background: #fff;">

<view class="img-a">

<view class="t-b">

您好，

<br />

欢迎使用美发预约App

</view>

</view>

<view class="login-view" style="">

<view class="t-login">

<form class="cl">

<view class="t-a">

<text class="txt">手机号</text>

<input type="number" name="phone" placeholder="请输入您的手机号" maxlength="11" v-model="phone" />

</view>

<view class="t-a">

<text class="txt">密码</text>

<input type="password" name="code" maxlength="18" placeholder="请输入您的密码" v-model="pwd" />

</view>

<button @tap="login()">登 录</button>

<view class="reg" @tap="reg()">注 册</view>

</form>

<view class="t-f">

<text @click="goto">忘记密码</text></br>

<text>—————— 第三方账号登录 ——————</text>

</view>

<view class="t-e cl">

<view class="t-g" @tap="wxLogin()">

<image src="../../static/images/login/wechat.png"></image>

</view>

<view class="t-g" @tap="zfbLogin()">

<image src="../../static/images/login/qq.png"></image>

</view>

</view>

</view>

</view>

</view>

</template>

<script>

export default {

data() {

return {

phone: '', //手机号码

pwd: '' //密码

};

},

onLoad() {},

methods: {

//当前登录按钮操作

login() {

var that = this;

if (!that.phone) {

uni.showToast({

title: '请输入您的手机号',

icon: 'none'

});

return;

}

if (!/^[1][3,4,5,7,8,9][0-9]{9}$/.test(that.phone)) {

uni.showToast({

title: '请输入正确手机号',

icon: 'none'

});

return;

}

if (!that.pwd) {

uni.showToast({

title: '请输入您的密码',

icon: 'none'

});

return;

}

uni.showToast({

title: '登录成功！',

icon: 'none'

});

uni.switchTab({

url: '../home/home'

})

},

//注册按钮点击

reg() {

uni.showToast({

title: '注册跳转',

icon: 'none'

});

},

//忘记密码点击

goto() {

uni.showToast({

title: "忘记密码跳转"

})

},

//第三方微信登录

wxLogin() {

uni.showToast({

title: '微信登录',

icon: 'none'

});

},

//第三方QQ登录

zfbLogin() {

uni.showToast({

title: 'QQ登录',

icon: 'none'

});

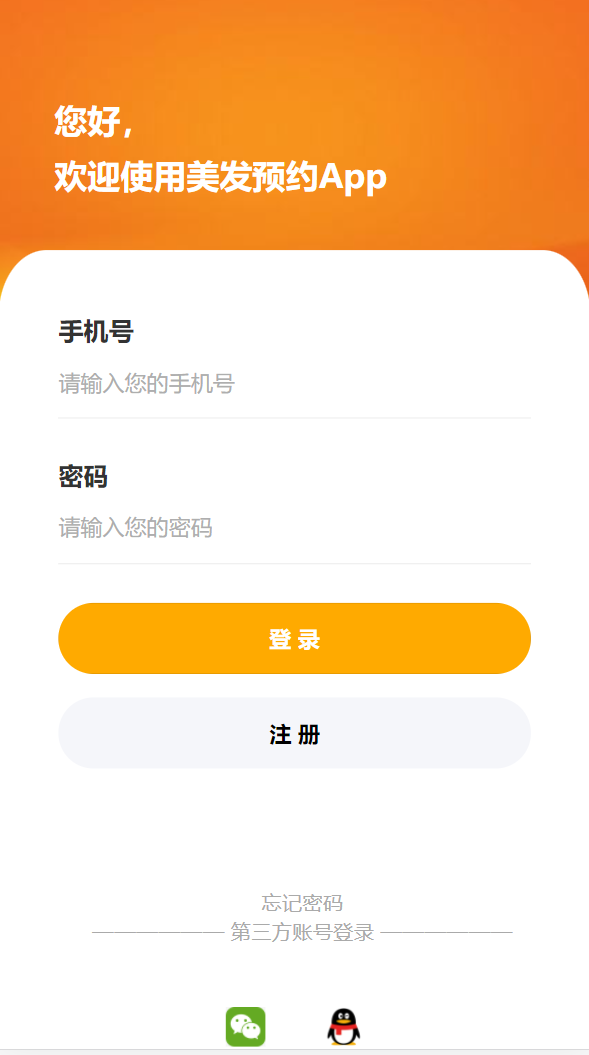
}

}

};

</script>

页面展示：



**3.2.1登陆页面**

### 3.2.2首页模块

实现思路：

在这一模块中我采用了模块化的思想，将每一部分内容设置成自定义的组件，并且最后在home.vue中引用各部分组件完成页面。在这一过程中大量的使用了uView组件库中的组件实现界面，如uni-nav-bar自定义顶部的导航栏，完成搜索框；使用u-swiper组件，并根据官方示例中的代码完成了轮播图的实现；u-grid宫格布局主要是添加了相关图标并设置了跳转事件实现了页面间的快速跳转页面下方则是我通过flex布局完成造型师和商品的展示。同时在这一模块中，我新建了一个“关于我们”界面，展示出美发店的一些信息，使APP更加符合用户习惯，方便用户了解信息。

关键代码:

home.vue:

<!-- 首页 -->

<template>

<view>

<!-- 导航栏 -->

<navbar></navbar>

<!-- 轮播图 -->

<wrap></wrap>

<!-- 通知栏 -->

<message></message>

<!-- 选项栏 -->

<selection></selection>

<!-- 首页造型师列表 -->

<people></people>

<!-- 首页商品列表 -->

<product></product>

</view>

</template>

<script>

import navbar from "../../components/home/u-nav.vue"

import wrap from "../../components/home/u-swiper.vue"

import message from "../../components/home/message.vue"

import selection from "../../components/home/selection.vue"

import people from "../../components/home/people.vue"

import product from "../../components/home/product.vue"

export default {

data() {

return {

}

},

onLoad: function() {},

components: {

navbar,

wrap,

message,

selection,

people,

product,

},

methods: {

}

}

</script>

顶部导航栏u-nav.vue:

<template>

<view class="navbar">

<uni-nav-bar color="#fe714b" background-color="#fe714b" @clickLeft="showCity" @clickRight="confirm()">

<block slot="left">

<view class="city">

店铺

<uni-icons type="arrowdown" color="#ffffff" size="22" />

</view>

</block>

<view class="input-view">

<uni-icons style="line-height: 30px;" type="search" size="22" color="#ffffff" />

<input confirm-type="search" class="nav-bar-input" type="text" placeholder="搜索服务/造型师" placeholder-class="c1"

@confirm="confirm">

</view>

<block slot="right">

<view class="search">

搜索

</view>

</block>

</uni-nav-bar>

</view>

</template>



**3.2.2.1顶部导航栏**

轮播图u-swiper.vue:

<template>

<view class="u-page">

<view class="u-demo-block">

<u-swiper

:list="list"

previousMargin="30"

nextMargin="30"

circular

:autoplay="true"

radius="5"

bgColor="#f0f0f0"

></u-swiper>

</view>

</view>

</template>

<script>

export default {

data() {

return {

current: 0,

currentNum: 0,

list: [

'../../static/images/swiper/01.jpeg',

'../../static/images/swiper/02.jpeg',

'../../static/images/swiper/03.jpeg ',

],

}

},

methods: {

change(e) {

// console.log('change', e);

},

click(e) {

console.log('click', e);

}

},

}



**3.2.2.2轮播图**

快速跳转selection.vue:

<template>

<view>

<u-line></u-line>

<u-grid :col="4" :border="false">

<u-grid-item @click="goto\_1">

<u-icon name="chat" :size="50" color="#ffaa00"></u-icon>

<view class="gird-text">关于我们</view>

</u-grid-item>

<u-grid-item @click="goto\_2">

<u-icon name="checkmark-circle" :size="45" color="#ffaa00"></u-icon>

<view class="gird-text">开始预约</view>

</u-grid-item>

<u-grid-item @click="goto\_3">

<u-icon name="shopping-cart" :size="55" color="#ffaa00"></u-icon>

<view class="gird-text">购物订单</view>

</u-grid-item>

<u-grid-item @click="goto\_4">

<u-badge count="3"></u-badge>

<u-icon name="file-text" :size="50" color="#ffaa00"></u-icon>

<view class="gird-text">预约订单</view>

</u-grid-item>

</u-grid>

<u-line></u-line>

</view>

</template>

<script>

export default {

data() {

return {

}

},

methods: {

goto\_1(){

uni.navigateTo({

url:"../../pages/about\_us/about\_us"

})

},

goto\_2(){

uni.switchTab({

url:"../../pages/service/service"

})

},

goto\_3(){

uni.navigateTo({

url:"../../pages/orderlist/orderlist"

})

},

goto\_4(){

uni.navigateTo({

url:"../../pages/appointment/appointment"

})

}

}

}

</script>



**3.2.2.3选择跳转**

通告栏message.vue:

<template>

<view>

<view class="message-box">

<view class="page-section swiper">

<view class="page-section-spacing">

<swiper style="height: 120rpx;" class="swiper" vertical="ture" circular="true"

indicator-dots='false' indicator-color="rgba(0,0,0,.0)" indicator-active-color="rgba(0,0,0,.0)"

autoplay="true" interval="4000">

<swiper-item class="swiper-list" v-for="(item, index) in messageData" :key="index">

<view class="message-tltle">{{item.title}}</view>

<view class="message-content">

{{item.content}}

</view>

</swiper-item>

</swiper>

</view>

</view>

</view>

</view>

</template>

<script>

export default {

data() {

return {

messageData: [{

title: '「太原1号店」12月31日起暂停营业',

content: '由于配合市政道路施工暂停营业'

},

{

title: '「太原2号店」新会员办卡享优惠',

content: '到店充值满300元返100元'

}

],

}

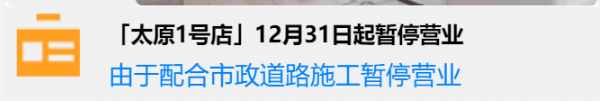
},

methods: {

}

}

</script>



**3.2.2.4通告栏**

关于我们about\_us.vue:

<!-- 关于我们 -->

<template>

<view>

<view class="location">

<image src="../../static/images/location.png" mode="widthFix"></image>

</view>

<u-card :title="title" :sub-title="subTitle">

<view class="" slot="body">

<view class="u-body-item u-flex u-border-bottom u-col-between u-p-t-0">

<view class="u-body-item-title u-line-2">引领美发时尚，捕捉造型流行趋势</view>

<image src="../../static/images/swiper/01.jpeg" mode="aspectFill"></image>

</view>

</view>

</u-card>

<u-card :title="title1" :sub-title="subTitle1">

<view class="" slot="body">

<view class="u-body-item u-flex u-border-bottom u-col-between u-p-t-0">

<view class="u-body-item-title u-line-2">

最顶尖的时尚发布、最贴心的造型设计、最合理的价格优势

</view>

<image src="../../static/images/swiper/02.jpeg" mode="aspectFill"></image>

</view>

</view>

</u-card>

</view>

</template>

<script>

export default {

data() {

return {

title: '太原一号店',

subTitle: '太原市万柏林区滨河东巷',

title1: '太原二号店',

subTitle1: '太原市迎泽区体育西路9号',

};

methods: {

};

}

}

</script>



**3.2.2.5关于我们**

页面展示：



**3.2.2首页页面**

### 3.2.3预约模块

实现思路：“”

该模块与实验课“修修哒”中的预约界面设计思路基本一致，仅仅是在部分内容中进行修改，使之更加符合本项目中需要符合的要求。如在用户需要选择造型师时，使用了uniapp的picker组件，设置出一个多列选择器，这么做的好处是用户可以选择不同店铺中的不同造型师。而在时间选择中，我使用了时间选择器的组件，用户可以在这一功能中完成日期和时间的选择。

关键代码:

预约界面service.vue:

<!-- 预约服务 -->

<template>

<view>

<!-- 用户信息 -->

<user></user>

<!-- 类别选择 -->

<type></type>

<!-- 造型师选择 -->

<person></person>

<!-- 时间选择 -->

<date></date>

<view class="loginForm\_dl">

<u-button @click="goto">立即下单</u-button>

</view>

</view>

</template>

<script>

import type from '../../components/service/type.vue'

import user from '../../components/service/user-info.vue'

import person from '../../components/service/person.vue'

import date from '../../components/service/date.vue'

export default {

data() {

return {

}

},

onLoad: function() {},

components: {

type,

user,

person,

date,

},

methods: {

goto(){

uni.navigateTo({

url:"../success/success"

})

}

}

}

</script>

<style>

.loginForm\_dl button {

background-color: #2C55C5;

color: #ffffff;

width: 50%;

height: 40px;

}

</style>

个人信息user-info.vue:

<template>

<view class="user">

<u-cell-group>

<u-cell-item title="账户余额:128.8" label="徐昕沛 18734660966"></u-cell-item>

</u-cell-group>

</view>

</template>



**3.2.3.1用户信息**

选择服务type.vue：

<template>

<view class="type">

<view class="piece-title">

<view>

<text class="tag">l</text>

<text class="piece-title-text">请选择您要预约的服务</text>

</view>

<view class="more">

<navigator>更多服务</navigator>

</view>

</view>

<view class="box">

<view v-for="item in testData" class="box-item" @click="dealChose(item)"

:class="item.id === curItem.id ? 'selected' : ''">

<img :src="item.icon" mode=""></img>

<text>{{item.msg}}</text>

</view>

</view>

</view>

</template>

<script>

export default {

data() {

return {

// 后端传递的数据

testData: [{

id: 1,

icon: "static/images/type/01.png",

msg: "染发"

},

{

id: 2,

icon: "static/images/type/02.png",

msg: "剪发"

},

{

id: 3,

icon: "static/images/type/03.png",

msg: "烫发"

},

{

id: 4,

icon: "static/images/type/04.png",

msg: "洗发"

},

],

curItem: {}

}

},

methods: {

dealChose(item) {

console.log(item);

// this.curItem.id = item.id;

this.$set(this.curItem, 'id', item.id);

}

}

}

</script>



**3.2.3.2服务选择**

选择造型师person.vue:

<template>

<view class="type">

<view class="piece-title">

<view>

<text class="tag">l</text>

<text class="piece-title-text">点击下方选择造型师和店铺</text>

</view>

</view>

<view>

<picker @change="bindPickerChange" :value="index" :range="array" range-key="name">

<view style="padding: 20px;background-color: white; text-align: center;">{{ array[index].name }}</view>

</picker>

</view>

</view>

</template>

<script>

export default {

data() {

return {

array: [{

name: '阿飞'

}, {

name: '阿布'

}, {

name: '阿福'

}, {

name: '阿健'

}],

index: 0,

}

},

methods: {

bindPickerChange: function(e) {

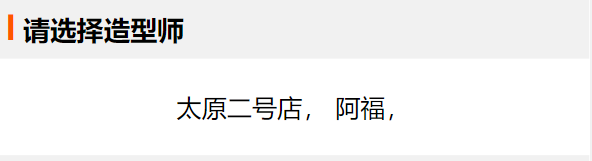
this.index = e.detail.value;

}

}

}

</script>



**3.2.3.3造型师选择**

选择日期时间date.vue：

<template>

<view class="page">

<view class="piece-title">

<view>

<text class="tag">l</text>

<text class="piece-title-text">点击下方选择日期时间</text>

</view>

</view>

<view class="example-body">

<uni-datetime-picker

type="datetime"

/>

</view>

</view>

</template>

<script>

export default {

data() {

return {

};

},

mounted() {

},

methods: {

change(e) {

this.single = e;

console.log("-change事件:", e);

},

},

};

</script>



**3.2.3.4日期时间选择**

预约成功界面success.vue:

<!-- 预约跳转界面 -->

<template>

<view>

<text class="h1">下单完成√</text>

<text class="h2">正在等待造型师确认订单</text>

<text class="h2">感谢您的耐心等候!</text>

<view class="loginForm\_dl">

<button type="default" @click="goto">返回首页</button>

</view>

</view>

</template>

<script>

export default {

data() {

return {

}

},

methods: {

goto() {

uni.switchTab({

url: "../home/home"

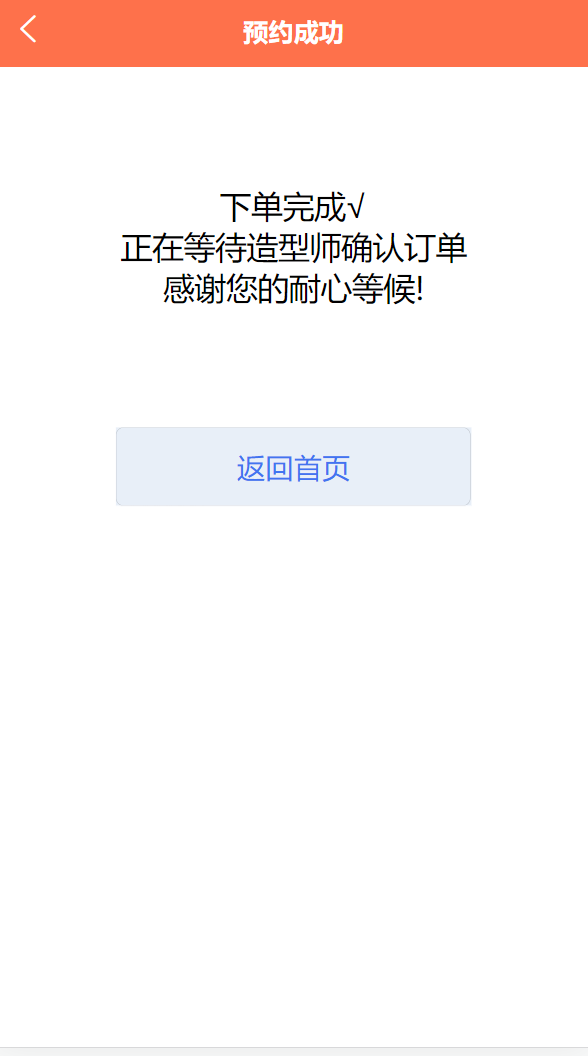
})

}

}

}

</script>



**3.2.3.5预约成功界面**

页面展示：



**3.2.3预约界面**

### 3.2.4购物车模块

实现思路：

本模块主要是实现购物车功能，在这一部分中主要是通过checkbox多项选择器完成用户对商品的选择，同时设置方法完成对商品的全部选择。对于商品的件数和价格，也是通过定义add、reduce、totalNum和totalPrice方法，完成对于商品的增加减少、总件数和总价的统计并最后在页面中进行一个展示。分别设置了“删除”、“批量操作”、“去结算”三种按钮并合理布局使之符合用户的操作习惯。

关键代码:

<template>

<view>

<view v-if="show==true">

<view class="goods-detail" v-for="(item,index) in goods" :key="index">

<view class="detail-left">

<view class="goods-left">

<checkbox-group @change="selected(item)">

<label>

<checkbox class="selected" color="#555555" :checked="checked" />

</label>

</checkbox-group>

<image :src="item.goodsImage" style="width: 150rpx;height: 140rpx;"></image>

</view>

<view class="size">

<text style="font-size: 25rpx;">品名:{{item.size}}</text>

<text class="goods-price">￥{{item.price}}/件</text>

</view>

</view>

<view class="detail-right">

<text class="subtract" @click="reduce(item)">-</text>

<text class="num">{{item.num}}</text>

<text @click="add(item)" class="add">+</text>

<u-button type="error" size="mini" @click="ddd">删除</u-button>

</view>

</view>

</view>

<u-button type="primary" size="mini" style="width: 80px; height: 30px;" @click="all">批量操作

</u-button>

<view class="end">

<view class="end-left">

<checkbox-group @change="selectgoods()">

<label>

<checkbox :checked="allchecked" />全选

</label>

</checkbox-group>

<view>

总计：<text style="color: #f00000;font-weight: bold;">￥ {{totalPrice}}</text>

</view>

</view>

<view class="end-right" @click="showAll">

结算({{totalNum}})

</view>

</view>

</view>

</template>

<script>

export default {

data() {

return {

show: true,

allchecked: true,

checked: true,

goods: [{

size: "宣谷生姜洗发水",

num: 1,

flag: true,

price: 68.00,

goodsImage: "../../static/images/product/1.webp",

},

{

size: "沪美生姜洗发水",

num: 1,

flag: true,

price: 129.00,

goodsImage: "../../static/images/product/2.webp",

},

{

size: "好迪洗发水/洗发露",

num: 1,

flag: true,

price: 39.90,

goodsImage: "../../static/images/product/3.webp",

},

{

size: "小菲洗发水",

num: 1,

flag: true,

price: 129.90,

goodsImage: "../../static/images/product/4.webp",

},

],

}

},

methods: {

goshopping() {

uni.navigateTo({

url: '../contact/contact'

})

},

change(e) {

console.log(e)

},

selected(item) {

item.flag = !item.flag

if (!item.flag) {

this.allchecked = false

} else {

const a = this.goods.forEach((item) => {

return item.flag === true

})

if (a) {

this.allchecked = true

} else {

this.allchecked = false

}

}

},

selectgoods() {

this.allchecked = !this.allchecked

if (this.allchecked) {

this.goods.map(item => {

this.checked = true

item.flag = true

})

} else {

this.checked = false

this.goods.map(item => {

item.flag = false

})

}

},

reduce(item) {

let num = item.num

if (num > 1) {

num -= 1

} else if (num = 1) {

uni.showToast({

title: "不能再减少了"

})

}

item.num = num

},

add(item) {

let num = item.num

item.num = num + 1

},

ddd() {

uni.showToast({

title: "删除"

})

},

all() {

uni.showToast({

title: "批量操作"

})

},

showAll() {

uni.showToast({

title: "去结算界面"

})

}

},

computed: {

totalNum() {

let totalNum = 0;

this.goods.map(item => {

item.flag ? totalNum += item.num : totalNum += 0

})

return totalNum

},

totalPrice() {

let totalPrice = 0;

this.goods.map(item => {

item.flag ? totalPrice += item.num \* item.price : totalPrice += 0

})

return totalPrice

},

}

}

</script>

页面展示：



**3.2.4购物车页面**

### 3.2.5我的模块

实现思路：

这一模块是我在本次系统中功能设计最为复杂的一个模块，首先还是和首页界面一样，将各个部分需要实现的功能组件化，并最后在user.vue中引用完成页面。该模块中优惠券以及地址选择我使用了uView中的官方页面并加以修改，主要是要注意页面之间的联系。订单界面分别设置了用户的预约订单和用户购买商品的订单，主要是在页面通过u-tabs-swiper完成顶部标签的切换，在通过宫格组件u-grid完成对每个标签的设置，这样做的好处是在后期需要修改时避免页面代码过于复杂。关于历史商品浏览记录我是通过引用官方u-scroll-list横向滚动列表实现组件实现，进一步增添该模块的功能。

关键代码:

用户信息up-background.vue:

<template>

<view class="upbackground">

<view class="content">

<view class="flex-a">

<view class="user\_img">

<u-avatar :src="src" shape="circle" size="60"></u-avatar>

</view>

<view class="user\_text">

徐昕沛</br>

已在美发预约App126天

</view>

<view class="icon-right">

<u-icon name="setting" color="#FFFFFF" size="30" @click="icon"></u-icon>

</view>

</view>

<view class="item">

<view class="tj-item">

<text class="num">128.8</text></br>

<text>会员卡余额</text>

</view>

<view class="tj-item"@click="coupon">

<text class="num">2</text></br>

<text>优惠券</text>

</view>

<view class="tj-item">

<text class="num">563.5</text></br>

<text>累计消费</text>

</view>

</view>

</view>

</view>

</view>

</template>

<script>

export default {

data() {

return {

src: '../../static/images/avatar.png'

}

},

methods: {

icon(){

uni.navigateTo({

url: '../../pages/set/set'

})

},

coupon(){

uni.navigateTo({

url:"../../pages/coupon/coupon"

})

}

}

}

</script>



**3.2.5.1用户信息**

设置界面set.vue：

<template>

<view class="container">

<u-cell-group>

<u-cell icon="integral-fill" title="个人资料" @click="person"></u-cell>

<u-cell icon="setting-fill" title="收货地址" @click="position"></u-cell>

<view class="cell">

<u-cell icon="chat-fill" title="消息推送"></u-cell>

<u-switch v-model="value13" asyncChange @change="asyncChange"></u-switch>

</view>

</u-cell-group>

<view class="loginForm\_dl">

<u-button @click="goto">点击退出登录</u-button>

</view>

</view>

</template>

<script>

export default {

data() {

return {

value13:true

};

},

methods: {

asyncChange(e) {

uni.showModal({

content: e ? '确定要打开吗' : '确定要关闭吗',

success: (res) => {

if (res.confirm) {

this.value13 = e

}

}

})

},

person(){

uni.showToast({

title:"跳转到个人资料"

})

},

position(){

uni.navigateTo({

url:"../position/position"

})

},

goto(){

uni.navigateTo({

url: "../login/login"

})

}

}

};

</script>



**3.2.5.2设置界面**

收货地址界面position.vue：

<template>

<view>

<view class="item" v-for="(res, index) in siteList" :key="res.id">

<view class="top">

<view class="name">{{ res.name }}</view>

<view class="phone">{{ res.phone }}</view>

<view class="tag">

<text v-for="(item, index) in res.tag" :key="index"

:class="{red:item.tagText=='默认'}">{{ item.tagText }}</text>

</view>

</view>

<view class="bottom">

{{ res.site }}

<u-icon name="edit-pen" :size="40" color="#999999"></u-icon>

</view>

</view>

<view class="addSite" @tap="toAddSite">

<view class="add">

<u-icon name="plus" color="#ffffff" class="icon" :size="30"></u-icon>新建收货地址

</view>

</view>

</view>

</template>

<script>

export default {

data() {

return {

siteList: []

};

},

onLoad() {

this.getData();

},

methods: {

getData() {

this.siteList = [{

id: 1,

name: '徐昕沛',

phone: '187\*\*\*\*0966',

tag: [{

tagText: '默认'

},

{

tagText: '家'

}

],

site: '山西省大同市魏都新城 魏都大道66号'

},

{

id: 2,

name: '徐昕沛',

phone: '157\*\*\*\*4893',

tag: [{

tagText: '学校'

}],

site: '山西省太原市中北大学 学院路3号'

},

];

},

toAddSite() {

uni.navigateTo({

url:"../addsite/addsite"

});

}

}

};

</script>



**3.2.5.3收货地址界面**

添加用户地址addsite.vue：

<template>

<view class="wrap">

<view class="top">

<view class="item">

<view class="left">收货人</view>

<input type="text" placeholder-class="line" placeholder="请填写收货人姓名" />

</view>

<view class="item">

<view class="left">手机号码</view>

<input type="text" placeholder-class="line" placeholder="请填写收货人手机号" />

</view>

<view class="item" @tap="showRegionPicker">

<view class="left">所在地区</view>

<input disabled type="text" placeholder-class="line" placeholder="省市区县、乡镇等" />

</view>

<view class="item address">

<view class="left">详细地址</view>

<textarea type="text" placeholder-class="line" placeholder="街道、楼牌等" />

</view>

<!-- <view class="site-clipboard">

<textarea placeholder-class="line" value="" placeholder="粘贴文本,可自动识别姓名和地址等" />

<view class="clipboard">

地址粘贴板

<u-icon name="arrow-down" class="icon" :size="20"></u-icon>

</view>

</view> -->

</view>

<view class="bottom">

<view class="tag">

<view class="left">标签</view>

<view class="right">

<text class="tags">家</text>

<text class="tags">公司</text>

<text class="tags">学校</text>

<view class="tags plus">

<u-icon size="22" name="plus"></u-icon>

</view>

</view>

</view>

<view class="default">

<view class="left">

<view class="set">设置默认地址</view>

<view class="tips">提醒：每次下单会默认推荐该地址</view>

</view>

<view class="right">

<switch color="red" @change="setDefault" />

</view>

</view>

</view>

<u-picker mode="region" ref="uPicker" v-model="show" />

<u-button type="error" @click="add">确认地址</u-button>

</view>

</template>

<script>

export default {

data() {

return {

show: false

};

},

methods: {

setDefault() {},

showRegionPicker() {

this.show = true;

},

add() {

uni.showToast({

title: "已添加地址"

})

}

}

};

</script>



**3.2.5.4添加用户地址界面**

优惠券界面coupon.vue:

<template>

<view class="u-wrap">

<view class="coupon">

<view class="content">

<view class="left">

<view class="sum">

￥

<text class="num">8</text>

</view>

<view class="type">抵用券</view>

</view>

<view class="centre">

<view class="title">【烫发】8元无门槛红包</view>

<view class="valid-date">今日到期</view>

</view>

<view class="right">

<view size="mini" class="immediate-use" :round="true" @click="use">立即使用</view>

</view>

</view>

</view>

<view class="coupon">

<view class="content">

<view class="left">

<view class="sum">

￥

<text class="num">6</text>

</view>

<view class="type">抵用券</view>

</view>

<view class="centre">

<view class="title">【剪发】6元无门槛红包</view>

<view class="valid-date">12月31日到期</view>

</view>

<view class="right">

<view size="mini" class="immediate-use" :round="true" @click="use">立即使用</view>

</view>

</view>

</view>

</view>

</template>

<script>

export default {

methods:{

use(){

uni.showToast({

title:"已使用"

})

}

}

};

</script>



**3.2.5.5优惠券界面**

订单宫格appointment.vue:

<template>

<view>

<u-line></u-line>

<view class="piece-title">

<view>

<text class="piece-title-text">预约订单</text>

</view>

<view class="more" open-type="redirect">

<navigator url="../../pages/appointment/appointment">

查看全部订单>

</navigator>

</view>

</view>

<u-grid :col="4" :border="false">

<u-grid-item>

<u-icon name="rmb-circle" :size="50" color="#ffaa00"></u-icon>

<view class="gird-text">待付款</view>

</u-grid-item>

<u-grid-item>

<u-icon name="clock" :size="50" color="#ffaa00"></u-icon>

<view class="gird-text">服务中</view>

</u-grid-item>

<u-grid-item>

<u-icon name="checkmark-circle" :size="50" color="#ffaa00"></u-icon>

<view class="gird-text">已完成</view>

</u-grid-item>

<u-grid-item>

<u-icon name="close-circle" :size="50" color="#ffaa00"></u-icon>

<view class="gird-text">已取消</view>

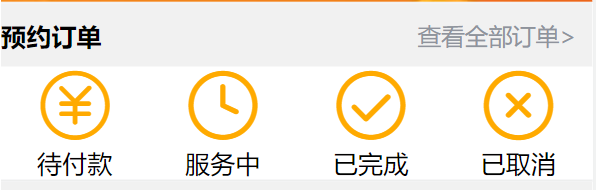
</u-grid-item>

</u-grid>

<u-line></u-line>

</view>

</template>



**3.2.5.6订单宫格**

预约订单信息界面appointment.vue:

<!-- 预约订单信息 -->

<template>

<view>

<view class="wrap">

<view class="u-tabs-box">

<u-tabs-swiper activeColor="#f29100" ref="tabs" :list="list" :current="current" @change="change" :is-scroll="false" swiperWidth="750"></u-tabs-swiper>

</view>

<swiper class="swiper-box" :current="swiperCurrent" @transition="transition" @animationfinish="animationfinish">

<swiper-item class="swiper-item">

<scroll-view scroll-y style="height: 100%;width: 100%;" @scrolltolower="reachBottom">

<view class="page-box">

<view class="order" v-for="(res, index) in orderList[0]" :key="res.id">

<view class="top">

<view class="left">

<u-icon name="home" :size="30" color="rgb(94,94,94)"></u-icon>

<view class="store">{{ res.store }}</view>

<u-icon name="arrow-right" color="rgb(203,203,203)" :size="26"></u-icon>

</view>

<view class="right">{{ res.deal }}</view>

</view>

<view class="item" v-for="(item, index) in res.goodsList" :key="index">

<view class="left"><image :src="item.goodsUrl" mode="aspectFill"></image></view>

<view class="content">

<view class="title u-line-2">{{ item.title }}</view>

<view class="type">{{ item.type }}</view>

<view class="delivery-time">预约时间 {{ item.deliveryTime }}</view>

</view>

<view class="right">

<view class="price">

￥{{ priceInt(item.price) }}

<text class="decimal">.{{ priceDecimal(item.price) }}</text>

</view>

</view>

</view>

<view class="bottom">

<view class="more"><u-icon name="more-dot-fill" color="rgb(203,203,203)"></u-icon></view>

<view class="logistics btn"></view>

<view class="exchange btn">退款</view>

<view class="evaluate btn">评价</view>

</view>

</view>

<u-loadmore :status="loadStatus[0]" bgColor="#f2f2f2"></u-loadmore>

</view>

</scroll-view>

</swiper-item>

<swiper-item class="swiper-item">

<scroll-view scroll-y style="height: 100%;width: 100%;" @scrolltolower="reachBottom">

<view class="page-box">

<view class="order" v-for="(res, index) in orderList[1]" :key="res.id">

<view class="top">

<view class="left">

<u-icon name="home" :size="30" color="rgb(94,94,94)"></u-icon>

<view class="store">{{ res.store }}</view>

<u-icon name="arrow-right" color="rgb(203,203,203)" :size="26"></u-icon>

</view>

<view class="right">{{ res.deal }}</view>

</view>

<view class="item" v-for="(item, index) in res.goodsList" :key="index">

<view class="left"><image :src="item.goodsUrl" mode="aspectFill"></image></view>

<view class="content">

<view class="title u-line-2">{{ item.title }}</view>

<view class="type">{{ item.type }}</view>

<view class="delivery-time">预约时间 {{ item.deliveryTime }}</view>

</view>

<view class="right">

<view class="price">

￥{{ priceInt(item.price) }}

<text class="decimal">.{{ priceDecimal(item.price) }}</text>

</view>

</view>

</view>

<view class="bottom">

<view class="more"><u-icon name="more-dot-fill" color="rgb(203,203,203)"></u-icon></view>

<view class="logistics btn"></view>

<view class="exchange btn">退款</view>

<view class="evaluate btn">评价</view>

</view>

</view>

<u-loadmore :status="loadStatus[1]" bgColor="#f2f2f2"></u-loadmore>

</view>

</scroll-view>

</swiper-item>

<swiper-item class="swiper-item">

<scroll-view scroll-y style="height: 100%;width: 100%;">

<view class="page-box">

<view>

<view class="centre">

<image src="https://cdn.uviewui.com/uview/template/taobao-order.png" mode=""></image>

<view class="explain">

您还没有相关的订单

<view class="tips">可以去看看有那些想买的</view>

</view>

<view class="btn">随便逛逛</view>

</view>

</view>

</view>

</scroll-view>

</swiper-item>

<swiper-item class="swiper-item">

<scroll-view scroll-y style="height: 100%;width: 100%;" @scrolltolower="reachBottom">

<view class="page-box">

<view class="order" v-for="(res, index) in orderList[3]" :key="res.id">

<view class="top">

<view class="left">

<u-icon name="home" :size="30" color="rgb(94,94,94)"></u-icon>

<view class="store">{{ res.store }}</view>

<u-icon name="arrow-right" color="rgb(203,203,203)" :size="26"></u-icon>

</view>

<view class="right">{{ res.deal }}</view>

</view>

<view class="item" v-for="(item, index) in res.goodsList" :key="index">

<view class="left"><image :src="item.goodsUrl" mode="aspectFill"></image></view>

<view class="content">

<view class="title u-line-2">{{ item.title }}</view>

<view class="type">{{ item.type }}</view>

<view class="delivery-time">预约时间 {{ item.deliveryTime }}</view>

</view>

<view class="right">

<view class="price">

￥{{ priceInt(item.price) }}

<text class="decimal">.{{ priceDecimal(item.price) }}</text>

</view>

</view>

</view>

<view class="bottom">

<view class="more"><u-icon name="more-dot-fill" color="rgb(203,203,203)"></u-icon></view>

<view class="logistics btn"></view>

<view class="exchange btn">删除订单</view>

<view class="evaluate btn">评价</view>

</view>

</view>

<u-loadmore :status="loadStatus[3]" bgColor="#f2f2f2"></u-loadmore>

</view>

</scroll-view>

</swiper-item>

</swiper>

</view>

</view>

</template>

<script>

export default {

data() {

return {

orderList: [[], [], [], []],

dataList: [

{

id: 1,

store: '太原一号店',

deal: '交易成功',

goodsList: [

{

goodsUrl: '../../static/images/person/person01.png',

title: '洗护一号套餐',

type: '造型师：阿飞',

deliveryTime: '2021-12-20 13:30',

price: '84.50',

},

]

},

{

id: 2,

store: '太原一号店',

deal: '交易失败',

goodsList: [

{

goodsUrl: '../../static/images/person/person02.png',

title: '剪发二号套餐',

type: '造型师:阿布',

deliveryTime: '2021-12-14 10:20',

price: '45.00',

}

]

},

{

id: 3,

store: '太原二号店',

deal: '交易失败',

goodsList: [

{

goodsUrl: '../../static/images/person/person03.png',

title: '染发二号套餐',

type: '造型师:阿福',

deliveryTime: '2021-12-17 17:10',

price: '162.00',

},

]

},

{

id: 4,

store: '太原二号店',

deal: '交易成功',

goodsList: [

{

goodsUrl: '../../static/images/person/person04.png',

title: '烫染一号套餐',

type: '造型师:阿健',

deliveryTime: '2021-12-22 19:10',

price: '153.50',

},

]

},

],

list: [

{

name: '待付款'

},

{

name: '服务中'

},

{

name: '已完成'

},

{

name: '已取消',

}

],

current: 0,

swiperCurrent: 0,

tabsHeight: 0,

dx: 0,

loadStatus: ['loadmore','loadmore','loadmore','loadmore'],

};

},

onLoad() {

this.getOrderList(0);

this.getOrderList(1);

this.getOrderList(3);

},

computed: {

// 价格小数

priceDecimal() {

return val => {

if (val !== parseInt(val)) return val.slice(-2);

else return '00';

};

},

// 价格整数

priceInt() {

return val => {

if (val !== parseInt(val)) return val.split('.')[0];

else return val;

};

}

},

methods: {

reachBottom() {

// 此tab为空数据

if(this.current != 2) {

this.loadStatus.splice(this.current,1,"loading")

setTimeout(() => {

this.getOrderList(this.current);

}, 1200);

}

},

// 页面数据

getOrderList(idx) {

for(let i = 0; i < 5; i++) {

let index = this.$u.random(0, this.dataList.length - 1);

let data = JSON.parse(JSON.stringify(this.dataList[index]));

data.id = this.$u.guid();

this.orderList[idx].push(data);

}

this.loadStatus.splice(this.current,1,"loadmore")

},

// tab栏切换

change(index) {

this.swiperCurrent = index;

this.getOrderList(index);

},

transition({ detail: { dx } }) {

this.$refs.tabs.setDx(dx);

},

animationfinish({ detail: { current } }) {

this.$refs.tabs.setFinishCurrent(current);

this.swiperCurrent = current;

this.current = current;

}

}

};

</script>



**3.2.5.7预约订单信息界面**

商品订单信息界面同预约订单信息界面

历史浏览商品history.vue:

<template>

<view class="u-page">

<view class="u-demo-block">

<u-line></u-line>

<view class="piece-title">

<view>

<text class="piece-title-text">历史浏览商品</text>

</view>

</view>

<u-scroll-list indicatorColor="#fff0f0" indicatorActiveColor="#f56c6c" @right="right" @left="left">

<view class="scroll-list" style="flex-direction: row;">

<view class="scroll-list\_\_goods-item" v-for="(item, index) in goodsArr" :key="index":class="[(index === 9) && 'scroll-list\_\_goods-item--no-margin-right']">

<image class="scroll-list\_\_goods-item\_\_image" :src="item.goodsUrl" mode="aspectFit"></image>

<text class="scroll-list\_\_goods-item\_\_text">￥{{ item.price }}</text>

</view>

<view class="scroll-list\_\_show-more" @tap="showMore">

<text class="scroll-list\_\_show-more\_\_text">查看更多</text>

<u-icon name="arrow-leftward" color="#f56c6c" size="12"></u-icon>

</view>

</view>

</u-scroll-list>

</view>

</view>

</template>

<script>

export default {

data() {

return {

goodsArr: [{

price: '68.00',

goodsUrl: '../../static/images/product/1.webp'

},

{

price: '129.00',

goodsUrl: '../../static/images/product/2.webp'

},

{

price: '39.90',

goodsUrl: '../../static/images/product/3.webp'

},

{

price: '129.90',

goodsUrl: '../../static/images/product/4.webp'

},

{

price: '29.9',

goodsUrl: '../../static/images/product/5.webp'

},

{

price: '19.8',

goodsUrl: '../../static/images/product/6.webp'

},

{

price: '19.00',

goodsUrl: '../../static/images/product/7.webp'

},

{

price: '112.00',

goodsUrl: '../../static/images/product/8.webp'

},

],

}

},

methods: {

showMore() {

uni.$u.toast('查看更多')

},

left() {

console.log('left');

},

right() {

console.log('right');

}

},

}

</script>



**3.2.5.8历史浏览商品**

页面展示：



**3.2.5我的界面**

# 第四章 APP打包

## 4.1 打包过程

### 4.1.1[打包为原生App（云端）](https://uniapp.dcloud.io/quickstart?id=%e6%89%93%e5%8c%85%e4%b8%ba%e5%8e%9f%e7%94%9fapp%ef%bc%88%e4%ba%91%e7%ab%af%ef%bc%89)

1在HBuilderX工具栏，点击发行，选择原生app-云端打包，如下图:

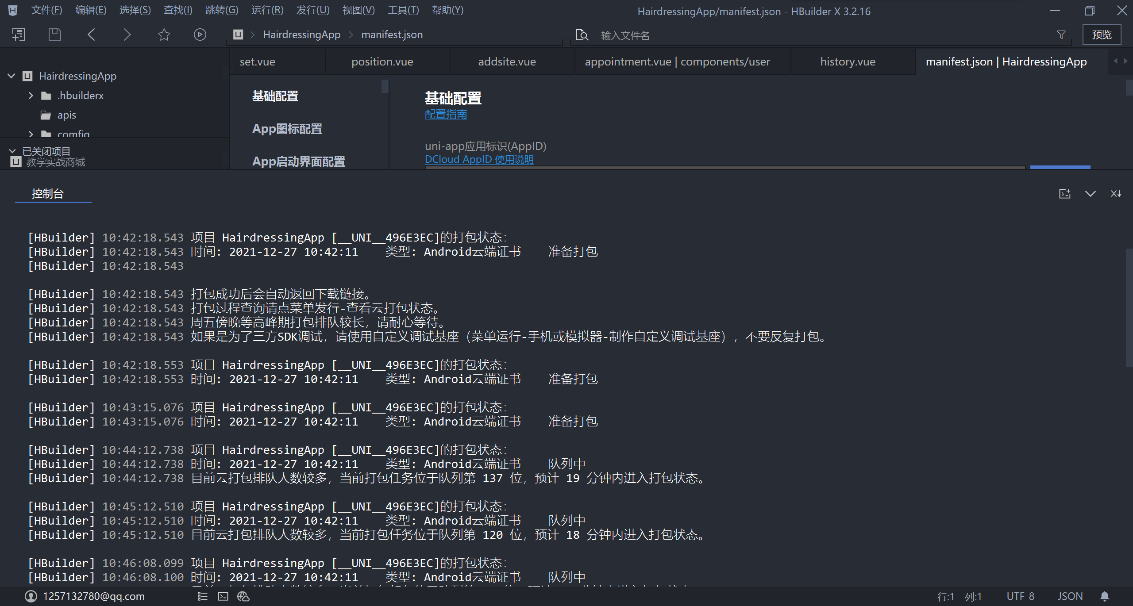
****

**4.1.1.1原生App云打包**

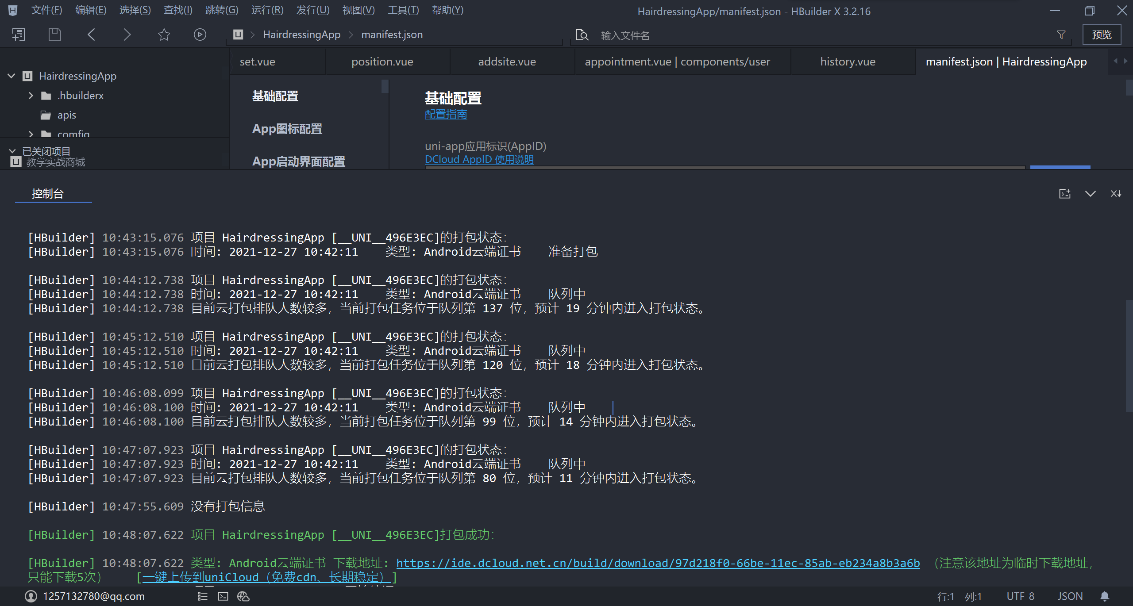
2选择证书,点击打包开始进行打包:



**4.1.1.2选择证书开始打包**



**4.1.1.3开始排队,等待打包**



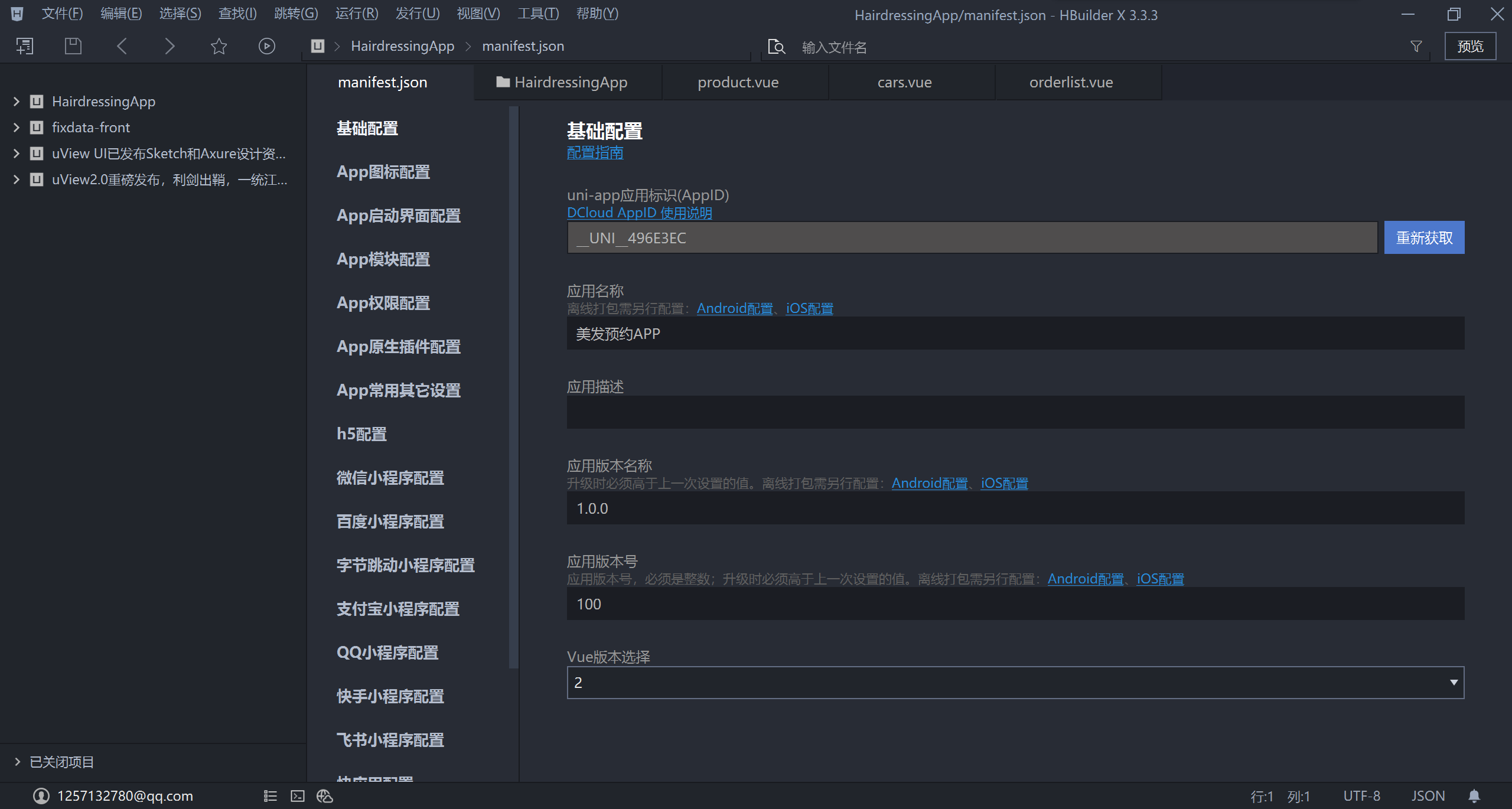
**4.1.1.4打包完成,生成下载链接**

3通过下载链接下载安装包,使用手机安装,完成对App的查看:

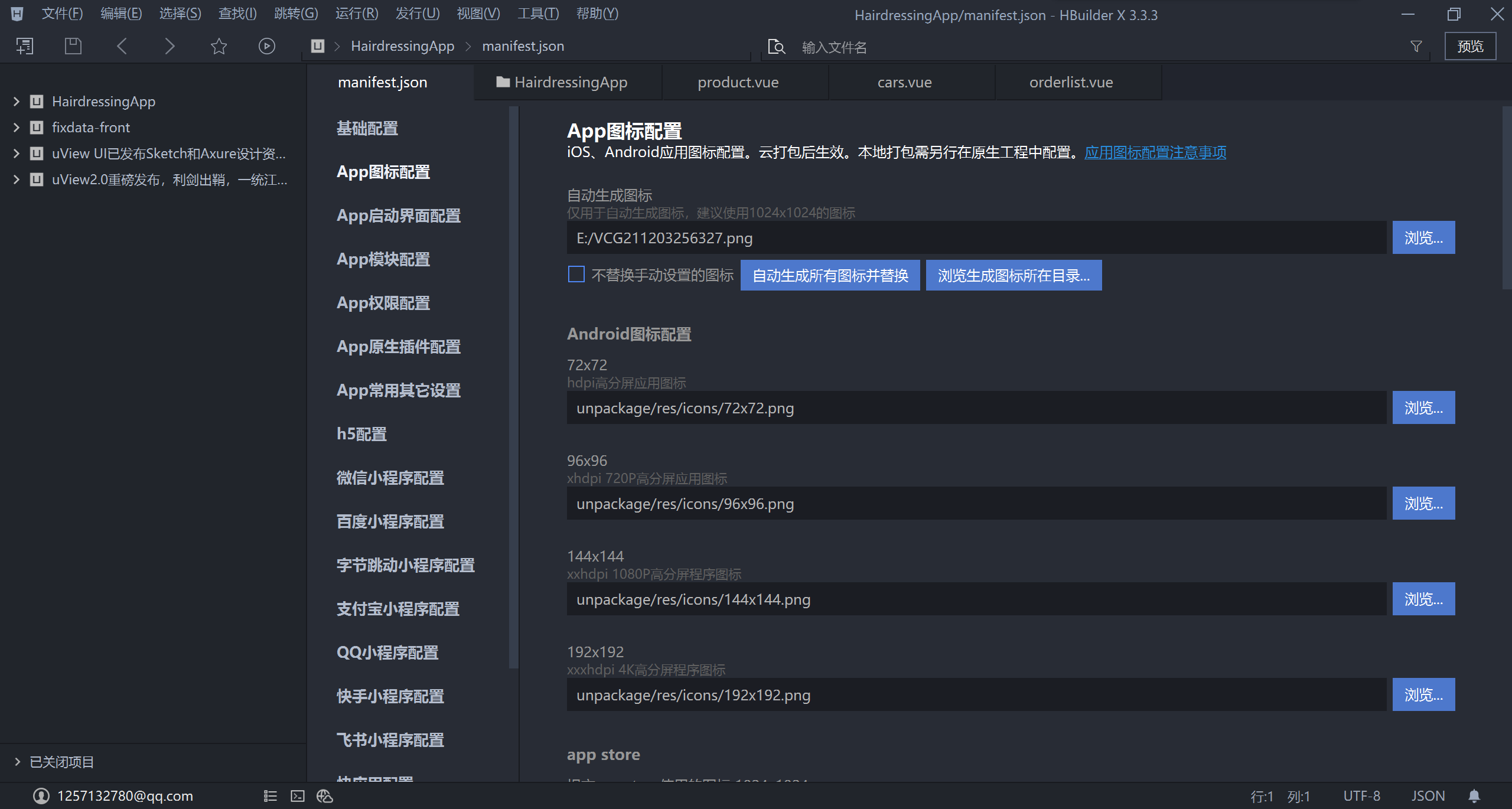


**4.1.1.5在手机上打开App**

## 4.2设计资源



**4.2.1基础配置**



**4.2.2App图标配置**