“一族一谱”管理系统

微信小程序技术文档

小组：软件工程第六组

组员：王聪 曾德龙 蔡楷欣 曾欣

程钰涵 陈冠旭 郑湘萍

版次： v4.2

日期：2021年4月20日

文档修改历史记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 修改人 | 日期 | 修改内容 | 备注 |
| v1.0 | 王聪 | 2021.4.20 | 添加UI框架说明和调查结果 | 初始内容 |
| V1.1 | 王聪 | 2021.4.21 | 添加UI规定内容 | 初始内容 |
| V1.2 | 王聪 | 2021.4.25 | 添加登录、版本兼容模块 | 初始内容 |
| V1.3 | 王聪 | 2021.4.26 | 修订 | 修正 |
| V2.0 | 陈冠旭 | 2021.4.28 | 添加消息发布 | 初始内容 |
| V3.0 | 曾德龙 | 2021.4.30 | 添加族谱树 | 初始内容 |
| V3.1 | 王聪、陈冠旭 | 2021.5.5 | 修改样式和错别字 | 修正内容 |
| V3.2 | 曾德龙 | 2021.5.10 | 数据库的补充寿命 | 补充说明 |
| V4.0 | 王聪 | 2021.5.15 | 增加1中ICON的说明，说明登陆的具体流程 | 增加 |
| V4.1 | 曾德龙、陈冠旭 | 2021.5.16 | 增加必要的新的文字说明 | 补充说明 |
| V4.2 | 曾德龙 | 2021.5.28 | 补充了申请信息的描述 | 补充说明 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 UI框架的选择 4](#_Toc75717076)

[1.1 可供选择的UI框架 4](#_Toc75717077)

[1.1.1 WeUI 4](#_Toc75717078)

[1.1.2 iview weapp 5](#_Toc75717079)

[1.1.3 ColorUI 5](#_Toc75717080)

[1.1.3 Wuss Weapp 6](#_Toc75717081)

[1.2 选择实现的框架 6](#_Toc75717082)

[1.2.1 ColorUI 7](#_Toc75717083)

[1.2.2 Wuss Weapp 7](#_Toc75717084)

[1.2.3 ColorUI和Wuss Weapp的选择 8](#_Toc75717085)

[2 UI的规定 10](#_Toc75717086)

[2.1 UI组件的选择 10](#_Toc75717087)

[2.2 ICON图标的选择 10](#_Toc75717088)

[2.3 十六进制颜色 11](#_Toc75717089)

[3 版本兼容性 12](#_Toc75717090)

[3.1 尽量兼容旧版本 12](#_Toc75717091)

[3.2 设定最低基础库版本 14](#_Toc75717092)

[4 登录模块的实现 16](#_Toc75717093)

[4.1 登录初始界面 16](#_Toc75717094)

[4.1.1 程序流程 16](#_Toc75717095)

[4.1.2 具体实现 16](#_Toc75717096)

[4.1.3 数据格式 19](#_Toc75717097)

[4.2 提出修改申请 19](#_Toc75717098)

[4.3 提出新增申请 20](#_Toc75717099)

[5 消息模块的实现 20](#_Toc75717100)

[5.1 消息发布模块 20](#_Toc75717101)

[5.2 历史消息查询模块 25](#_Toc75717102)

[5.3 最新历史消息滚动展示模块 28](#_Toc75717103)

[6 首页模块 30](#_Toc75717104)

[6.1族谱树模块的实现 30](#_Toc75717105)

[6.2统计模块的实现 32](#_Toc75717106)

[6.3关系模块的实现 34](#_Toc75717107)

[7成册模块 35](#_Toc75717108)

[1.2 项目范围 37](#_Toc75717109)

[1.3 文档概述 37](#_Toc75717110)

[1.3.1 文档相关内容 37](#_Toc75717111)

[1.3.2 文档参考 37](#_Toc75717112)

1 UI框架的选择

1.1 可供选择的UI框架

1.1.1 WeUI

WeUI 是一套同微信原生视觉体验一致的基础样式库，由微信官方设计团队为微信 Web 开发量身设计，可以令用户的使用感知更加统一。包含button、cell、dialog、 progress、 toast、article、actionsheet、icon等各式元素。GitHub 地址：<https://github.com/Tencent/weui> 效果：<https://weui.io> 开发文档参考：<https://github.com/Tencent/weui>



图1-1 WeUI主界面

WeUI 使用简单，风格微信原生风格，以绿色为主色 ，主要是两个大版本 ，1.x版本和2.x版本 ，相比1.x版本来说，2.x版本视觉上更好。

WeUI 使用第一个问题 ，文档不是很详细 ，访问 https://weui.io 能看到效果图，没有直观的显示效果对应的代码 ，从开发角度来说，文档不够直接，使用不是很方便。WeUI 使用第二个问题是：风格与微信视觉体验一致，在开发中我们需要更丰富漂亮的UI效果 ，需要修改WeUI的样式或自定义css，在实际开发中不是很便捷。

1.1.2 iview weapp

iView Weapp是由 TalkingData 发布的组件库，一套高质量的 微信小程序 UI 组件库。 GitHub 地址：<https://github.com/TalkingData/iview-weapp> 开发文档：<https://weapp.iviewui.com/docs/guide/start>



图1-2 iView Weapp界面介绍

优点：提供了丰富的UI组件，尤其是 tabBar、Tabs、抽屉、index索引 ，使用比较方便。

缺点：页面风格比较固定、修改起来比较困难。输入框在模拟器下经常无法输入，给应用调试带来了麻烦。

1.1.3 ColorUI

鲜亮的高饱和色彩，专注视觉的小程序组件库 GitHub 地址：<https://github.com/weilanwl/ColorUI> 开发文档参考（编辑中）：<https://www.color-ui.com>



图1-3 ColorUI界面介绍

ColorUI是一个Css类的UI组件库,相比于同类小程序组件库，ColorUI更注重于视觉交互，其组件在美观性方面比较突出。视觉效果漂亮，可针对自己的需求，直接修改或覆盖他的css样式，后面我们小组就尝试了该框架。

1.1.3 Wuss Weapp

Wuss Weapp是一款高质量，组件齐全，高自定义的微信小程序UI组件库。该框架的主要的优点是界面整洁美观，这点和我们的用户群体是比较相符合的。因为我们的用户大多数为中年人、或者是老年人，所以我们需要让界面尽可能的整洁美观。



图1-4 Wuss Weapp界面介绍

该框架的优点很明显，就是简单、可靠、易上手，开发出的微信小程序容易使用，整体简洁、美观。因此，不用写很多的CSS样式，也能开发出比较美观的微信小程序。但是该框架的缺点也比较明显，有的组件并没有需要自己去开发，或者修改源代码中的组件。

1.2 选择实现的框架

我们小组决定选择1.1小节中的ColorUI和Wuss Weapp进行作为微信小程序的UI框架进行开发，因为两种框架基本上不需要写很多的CSS样式，就可以实现基本的功能和样式。

1.2.1 ColorUI

经过讨论，我们小组决定先使用ColorUI作为微信小程序的UI框架。因为该框架颜值比较高，而且不用写很多的CSS样式。经过一段时间的开发，我们对登录界面的设计如图1-5所示：



图1-5基于ColorUI和Wuss Weapp的登陆界面

该登录页面设计的还算比较成功，整体比较美观。但是经过组内讨论认为用户（中年人、老年人）可能觉得比较花，于是我们经过仔细思考开发基于不同UI框架的登录界面，并进行对比和问卷调查。

1.2.2 Wuss Weapp

如图1-5所示，是我们小组使用Wuss Weapp进行开发的登陆界面。该界面的优势也很明显，整体比较简洁，很容易让用户抓住重点。

经过问卷调查发现，我们的主要用户对第二种的反馈比较好，普遍认为该框架具有整洁、醒目的特点。详细的介绍可以查看第1.2.3小节中的介绍。

1.2.3 ColorUI和Wuss Weapp的选择

我们小组发起了对微信小程序页面样式的偏好调查，该问卷主要是基于我们的用户群体（中年人、老年人）

以下是我们小组对微信小程序的UI框架的选择的问卷调查，本次问卷有效人数为200人，我们将年龄段分为15-25、26-35、36-45、46-60和61-100五个年龄段，可视化图如图1-6所示：

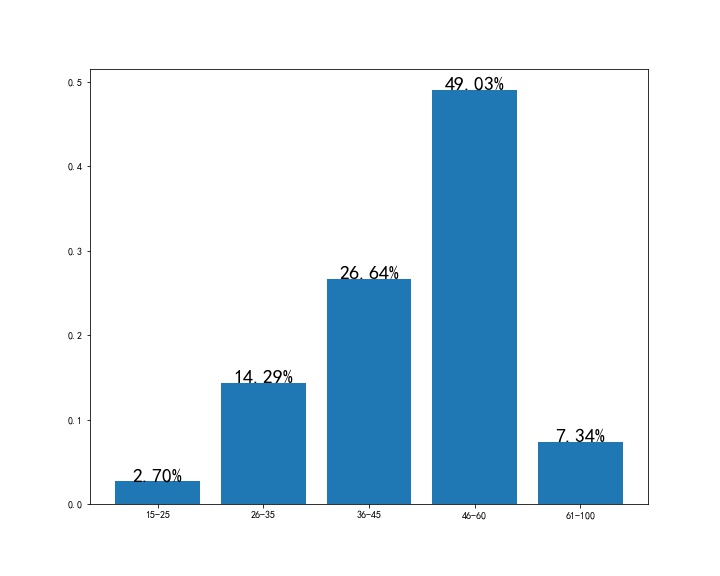


图1-6 填写问卷的年龄段

图1-7是选择ColorUI和Wuss Weapp人数分布，可以看到选择Wuss Weapp框架的人数比ColorUI框架的人数要多，比例为约为1.8：1。

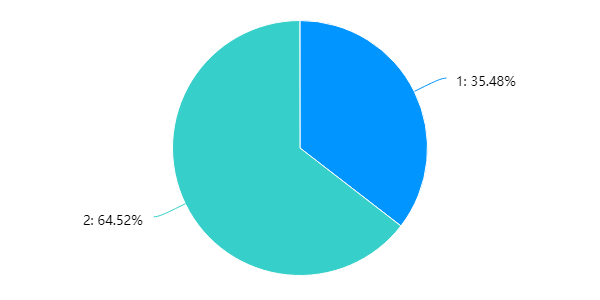


图1-7 选择1（ColorUI）和2（Wuss Weapp）的人数对比

S

我们的问卷统计得到，大部分的用户觉得该框架具有醒目、整洁、美观的特点，如图1-8所示。

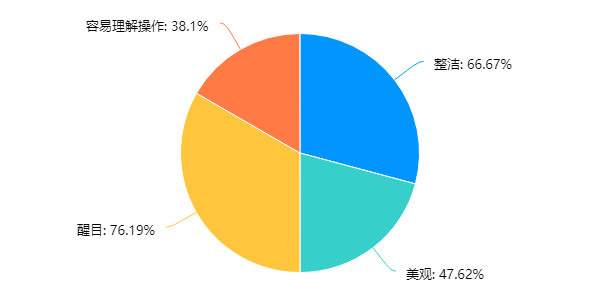


图1-8 Wuss Weapp的优点

实际上，我们组内也认为Wuss Weapp框架要由于ColorUI，因为该框架从技术上来说比较适合本次的开发，可以从源代码出进行修改。而且，我们组内也认为该框架具有醒目、美观的特点。所以，我们小组选择Wuss Weapp作为本次项目的微信小程序所使用到的框架。

2 UI的规定

为了保证小组中使用的UI是完全一样的，对下面几点内容进行规定：

2.1 UI组件的选择

微信小程序中的组件我们主要是使用Wuss Weapp中的组件，可以进行样式、跳转等自定义的修改。

例如在消息发布的UI我们就是用Wuss Weapp中的通知UI，如下图所示：



图 2-1 消息通知的UI组件 图 2-2 表单组件的UI组件

对于表单信息，我们也是直接使用Wuss Weapp中的表单组件，对接口进行适当的修改即可，如上图所示。

2.2 ICON图标的选择

对于图标ICON的选择，我们要求先到[iconfont-阿里巴巴矢量图标库](https://www.iconfont.cn/)中寻找合适的图标。例如我们想要搜索“我的”图标，如下图所示：

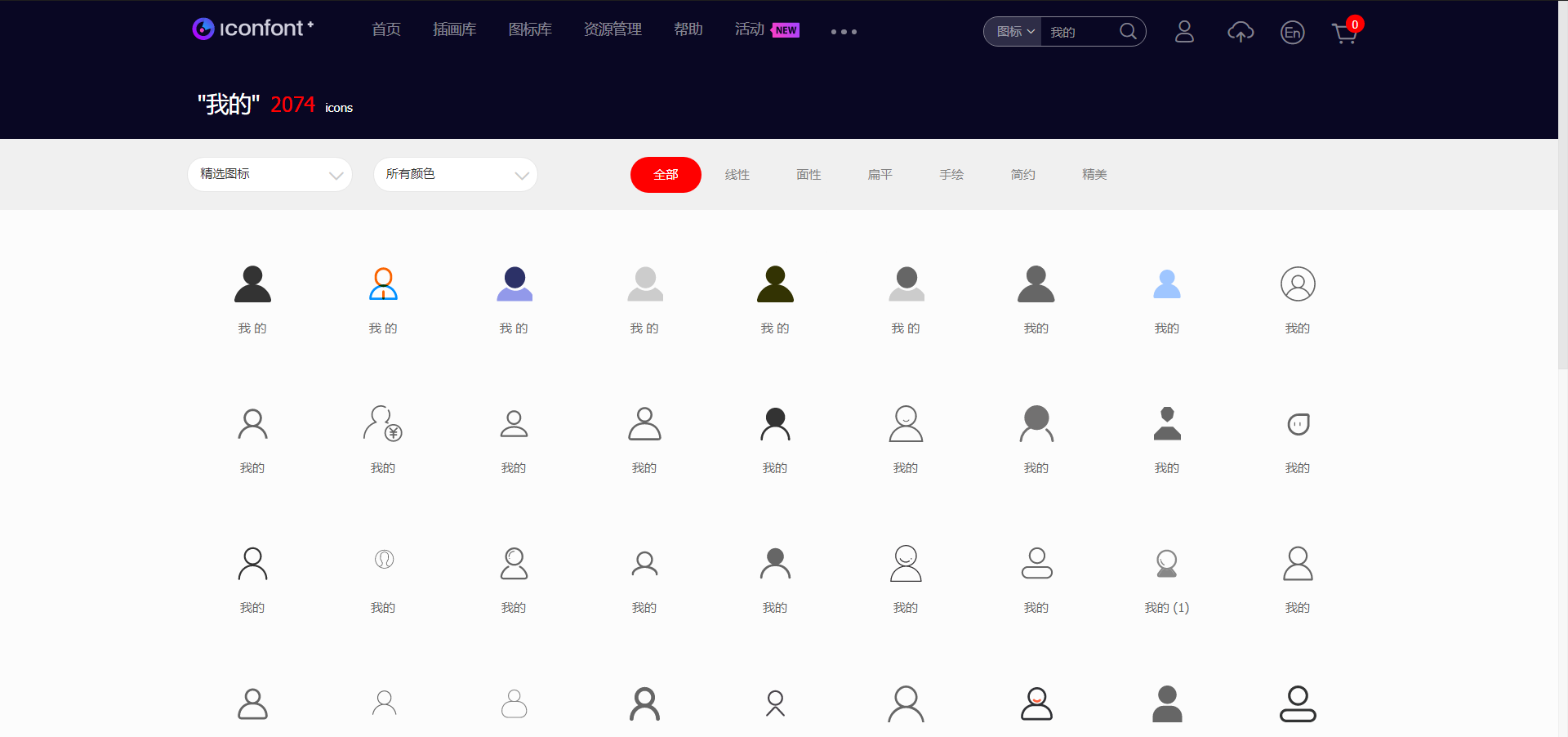


图 2-3 iconfont阿里巴巴矢量图标库

2.3 十六进制颜色

为了保持色彩和风格的统一，我们规定了所有的UI组件用到的十六进制颜色表。



图 2-4 十六进制颜色表

这样我们就实现了组员在开发的时候，使用到的颜色保持一致，有利于微信小程序风格的统一。

3 版本兼容性

为了考虑微信小程序的版本兼容性问题，我们在调用微信小程序的最新API时，从以下两个方面进行考虑：

3.1 尽量兼容旧版本

我们会考虑到兼容以前的微信小程序版本，从而在一些API的设定上会考虑旧版本的API，例如微信小程序的2.10.4版本才能够支函数wx.getUserProfile，新用户将会自动获取用户的头像和昵称，而旧版本用户我们将提示使用按钮，才能获取微信头像和昵称。如下所示，我们现在Load函数中判断是否可以使用wx.getUserProfile函数，如果可以就可以调用该函数：

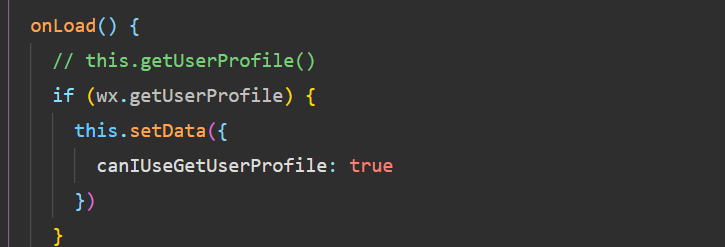


图 3-1 OnLoad中的判断

以下是getUserProfile函数的具体内容：

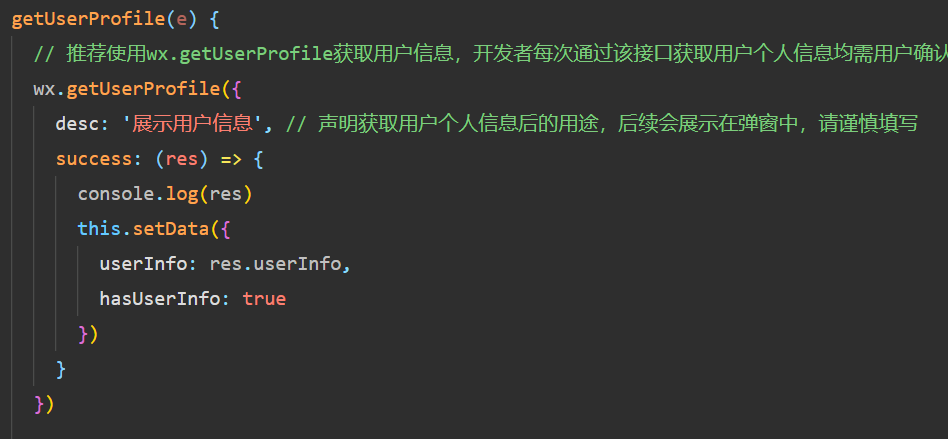


图 3-2 getUserProfile函数

如果用户的版本过低将会提示点击按钮之后获取头像等信息，调用的函数如下所示：

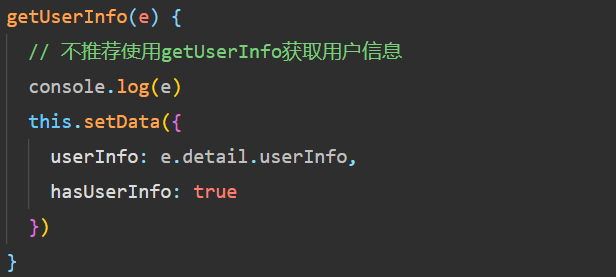


图 3-3 getUserInfo函数

3.2 设定最低基础库版本

为了解决低版本基础库无法兼容小程序的新功能的问题，我们设置了小程序最低基础库版本要求。通过查看近 30 天内访问当前小程序的用户所使用的基础库版本占比，了解当前用户使用的情况，从而设定微信小程序的最低基础库版本。如下图所示：



图 3-1 设置最低基础库版本

具体步骤为：登录小程序管理后台，进入「设置 - 基本设置 - 基础库最低版本设置」进行配置。

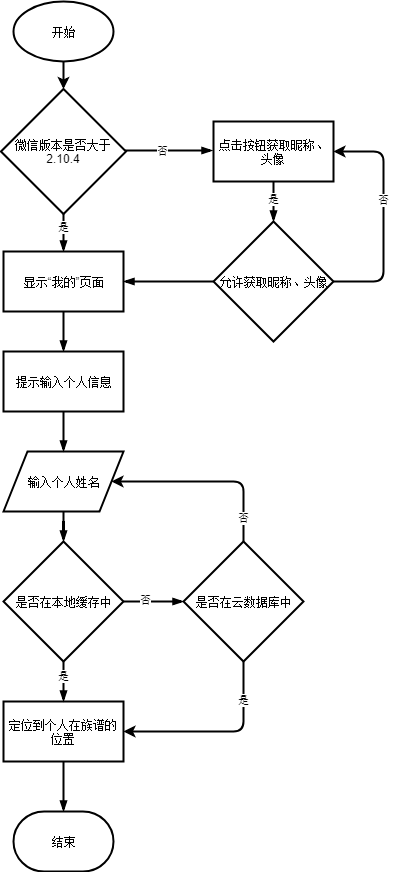
如果用户的版本号低于我们设定的版本号将会出现以下提示：



图 3-2 微信小程序的版本过低

4 登录模块的实现

4.1 登录初始界面

4.1.1 程序流程

微信小程序的“登录”流程图如右图所示。该模块的主要流程如下所示：

（1）判断用户的微信版本是否大于2.10.4，如果大于或等于登陆界面可以自动显示该用户的微信头像和昵称；如果微信版本小于该版本，用户需要点击按钮才能获取到微信的头像和昵称。

（2）接着将会判断用户是否是第一次登入“一族一谱”微信小程序。如果是第一次登入的话，进入（3）；否则转入（4）;

（3）用户第一次登入微信小程序中，提示并要求用户输入自己的姓名。此步骤中，向数据库查询用户输入的姓名相关的姓名列表，从而帮助用户更快的定位到用户所在的族谱位置。

（4）用户不是第一次登入微信小程序，在本地缓存中查询到用户的相关信息，从而定位到用户所在的族谱位置。

图 3-1 “登录”程序流程图

4.1.2 具体实现

和上图3-1中的“登录”程序流程图中表示的一样，微信小程序将会先尝试获取用户的微信昵称和头像。如果用户的微信版本大于2.10.4的话，将会自动获取用户的昵称和头像；如果微信小程序的版本小于2.10.4的话，用户需要点击按钮才能获取用户的昵称和头像。技术原理主要是通过微信中的wx.getUserProfile函数，如果微信小程序的版本大于2.10.4的话，就可以访问该接口。实现代码如下所示：

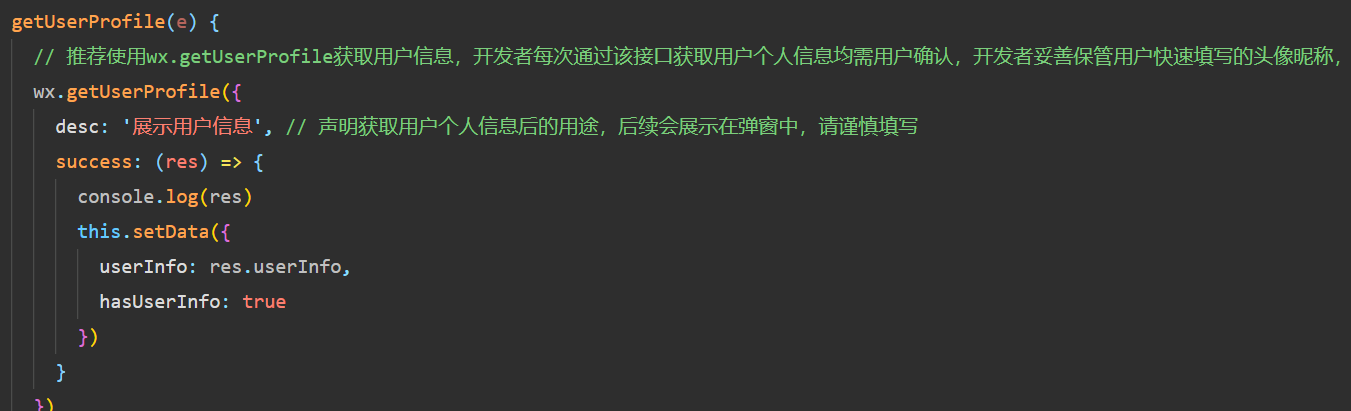


图 3-2 wx.getUserProfile函数

在个人的初始界面，如果是第一次登陆的话，微信小程序将会提示“请先完善个人信息”，如下图所示。



图 3-3 个人界面的提示

上述实现主要是通过函数wx.getStorage进行判断，查询微信小程序中的缓存中是否存在用户的缓存信息。如果存在缓存信息的话，将缓存信息保存，并定位到用户所在的族谱位置；如果不在将会调用自定义函数myShowModal，提示用户“请先完善个人信息”。

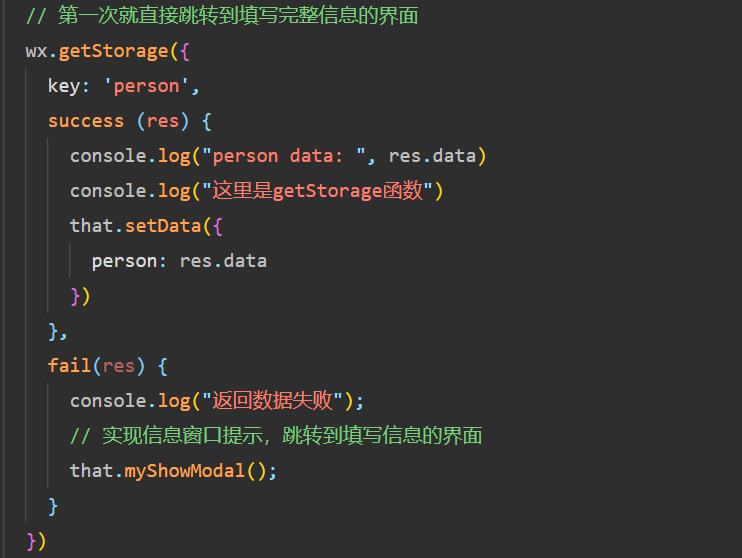


图 3-4 wx.getStorage函数

自定义函数myShowModal如下所示，只允许用户点击确定，然后跳转到填写信息的界面中：

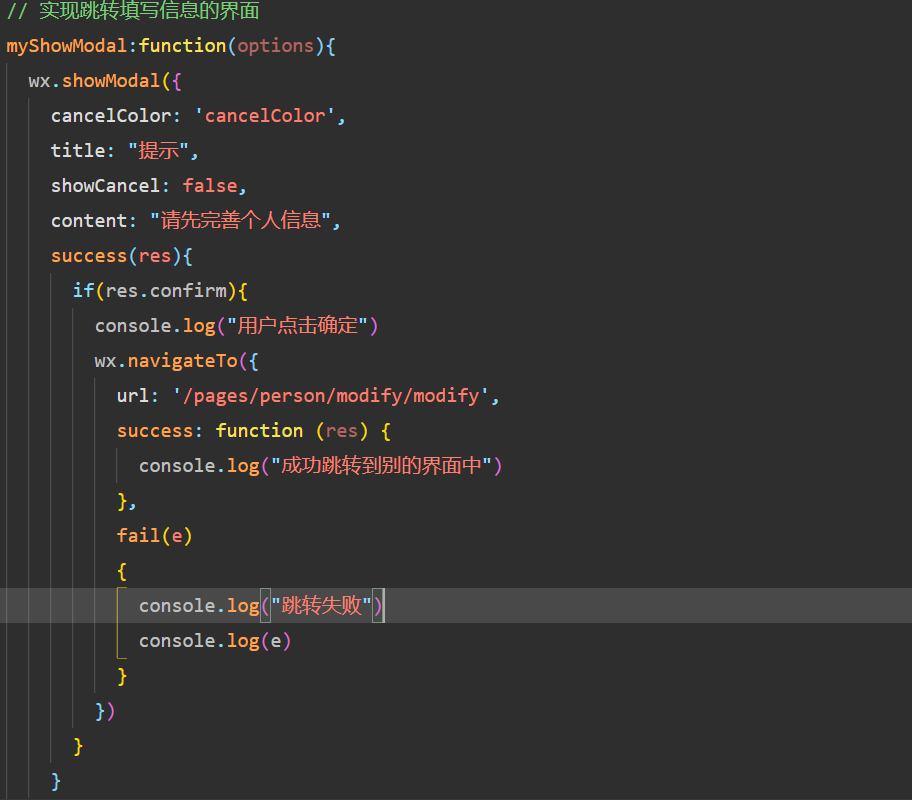


图 3-5 myShowModal函数

4.1.3 数据格式

在登陆初始界面中主要使用到的数据格式如下所示：

表3-1 信息录入表Person

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 默认值 | 注释 |
| person | String | 无 | 唯一识别id（自动生成） |
| userInfo | object | None | 用户的个人信息（头像等） |
| hasUserInfo | Boolean | false | 是否已经返回用户的昵称等 |
| canIUse | Boolean | false | 是否能够调用getUserInfo |
| canIUseGetUserProfile | Boolean | false | 是否能够获取用户信息 |
| canIUseOpenData | Boolean | false | 是否可以使用公共数据 |

4.2 提出修改申请

修改申请根据我们的定义通过设定操作符为0实现的，这里面最重要的是对用户的输入进行一定判断，防止一些异常情况出现。主要包括：

1. 姓名的长度判断，直接判断输入的字符串长度即可进行条件判断即可
2. 如果要修改性别，那需要考虑性别的更改是否会对婚配产生影响，比如已结婚的人是不能更改性别的，因为这样会出现同性婚姻的情况。
3. 出生日期的修改，主要是父与子之间的关系，需要获取父亲的生日，检查修改后父子的年龄差，如果父亲年龄减去儿子年龄小于12岁我们认定为不合理。
4. 检查信息是否有更改，如果用户信息没填，那意味着没修改，提交申请就多余了，就算用户有填写，也要检查与之前的信息是否一致，如果一致也没有修改

该部分用到的技术并不多，都是对字符串的条件判断。

4.3 提出新增申请

新增请求主体继承了修改请求，对各个信息的判断是一致的，但是多了一些其他判断：

1. 用户是否所有信息都有填写
2. 可以选择用户的关系，如果新增关系不是子女，不必考虑年龄问题

同时在这里我们用了w-validate里面内置了很多填写规范的属性，直接调用即可完成对应判断。

5 消息模块的实现

5.1 消息发布模块

该模块的UI设计是基于Wuss Weapp实现的，代码实现放在editAction模块。js文件中变量和函数的设计上如下：

表1 发布消息模块变量记录表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 默认值 | 注释 |
| title | string |  | 标题 |
| mainBody | string |  | 正文 |
| place | string |  | 地点 |
| undertaker | string |  | 举办方 |
| startTime | string | (当前北京时间如2021年05月30日 09:53) | 起始时间 |
| endTime | string | (当前北京时间如2021年05月30日 09:53) | 结束时间 |
| addZeroForMonth | string |  | 时间字符串的月份数补零到两位 |
| addZeroForDay | string |  | 时间字符串的日期数补零到两位 |
| addZeroForHour | string |  | 时间字符串的小时数补零到两位 |
| addZeroForMin | string |  | 时间字符串的分钟数补零到两位 |
| startYear | string | 2000 | 时间选择器可以设置的最小年份 |
| endYear | string | 2050 | 时间选择器可以设置的最大年份 |
| checkEmpty | bool | false | 检查输入是否有为空的情况 |

表2 函数记录表

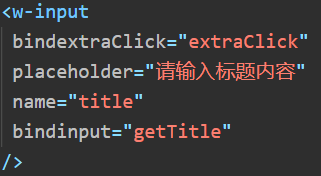
|  |  |
| --- | --- |
| 函数名 | 注释 |
| getMain | 获取输入的正文内容 |
| getTitle | 获取输入的标题 |
| getPlace | 获取输入的地点 |
| getUndertaker | 获取输入的举办方 |
| submitData | 检查输入是否合法，若合法将数据提交到Bomb数据库的表中 |
| onLoad | 加载页面，获取当前北京时间（包括年月日时分） |
| history | 进入查看历史信息页面 |
| changeDateTime | 设置起始时间 |
| changeDateTime1 | 设置结束时间 |
| changeDateTimeColumn | 起始时间的时间选择器调整 |
| changeDateTimeColumn1 | 结束时间的时间选择器调整 |

对于wxml文件的处理如下：

①页面的标题显示内容



②标题的输入框



我们在这里绑定js文件中的getTitle函数，输入的标题将会被实时获取，文本框内默认内容为“请输入标题内容”

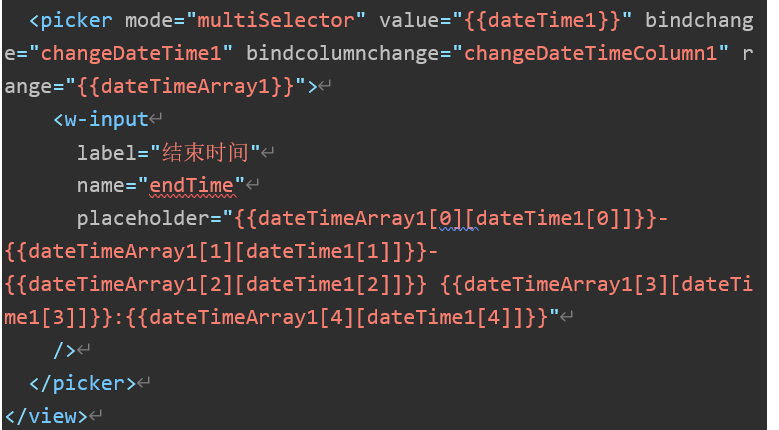
③正文的输入框



在这里我们限制正文输入字数200，输入框的样式使用textarea，绑定好js文件的getMain函数来实时获取输入的正文内容，placeholder表示了文本框内默认内容。

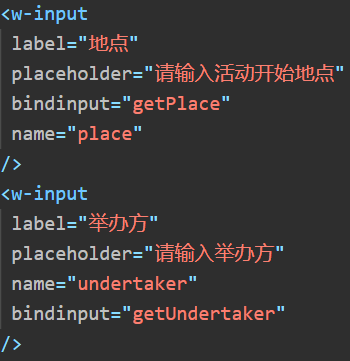
④起始时间和结束时间（时间选择器）输入框





picker模块由utils/dateTimePicker.js引入，在这个模块内我们可以将输入内容变为时间选择器，输入框内默认内容都是当前的北京时间，必须进行输入才能有相应的内容。

⑤地点和举办方输入框输入



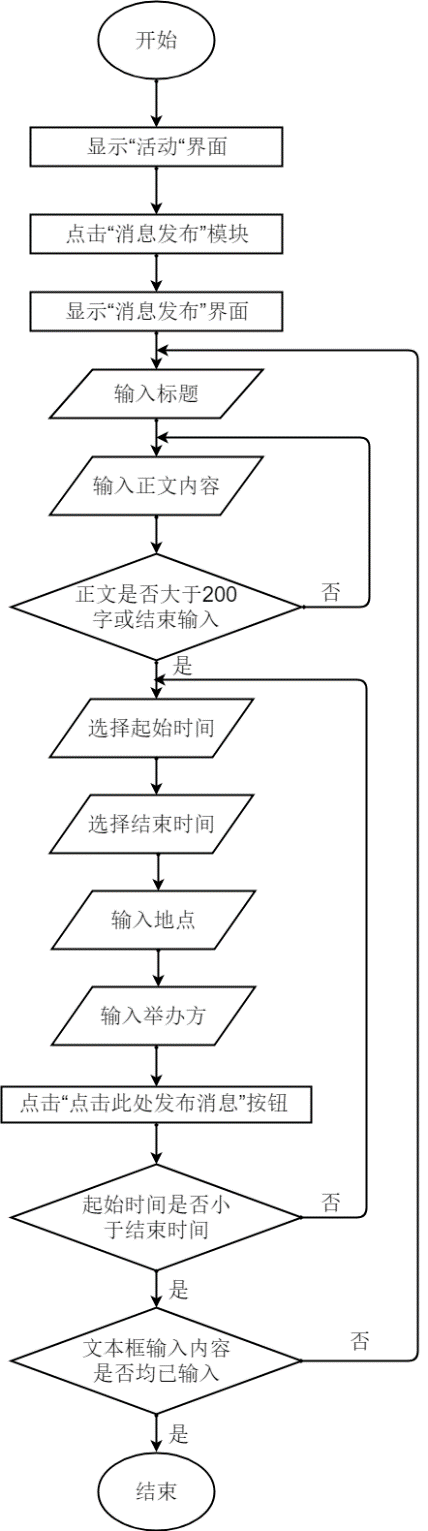
这两个输入框分别绑定js文件中的getPlace和getUndertaker函数，用于实时获取地点和举办方的输入。

⑥消息发布按钮



我们使用w-button组件，然后绑定好js文件中用于提交输入数据的submitData函数，点击按钮后会检查输入合法性，输入合法即可以发布消息。

程序流程图如下：



(1)点击活动模块，我们可以看到活动的界面，然后点击消息发布模块，就可以进入消息发布的界面

(2)在消息发布界面中，我们需要依次输入标题，正文，起始时间，结束时间，地点和举办方。正文处存在条件判断，如果填写字数小于等于200字可以继续输入，否则将无法输入。消息填写之后我们需要点击“点击此处发布消息按钮”。

(3)点击该按钮时会触发条件判断，首先会判断起始时间和结束时间的相对大小关系，在这里我们可以直接使用字符串比较的方法，因为接收到的起始和结束时间是经过处理的标准化字符串数据，可以直接使用运算符来进行比较。如果条件不满足则需要重新修改时间，条件满足进入下一步。

(4)判断完时间大小关系后需要判断是否存在输入为空的情况。如果存在则需要完善信息，如果输入均不为空，则可以将要发布的消息保存在数据库中。后续消息展示模块会将数据库的内容进行拉取并进行展示。发布消息的流程到这里结束

图1 消息发布模块流程图

5.2 历史消息查询模块

该模块的UI设计同样是基于Wuss Weapp实现的，简要消息展示的代码实现放在actionList模块，具体消息展示的代码实现放在actionDetail模块。js文件中变量和函数的设计上如下：

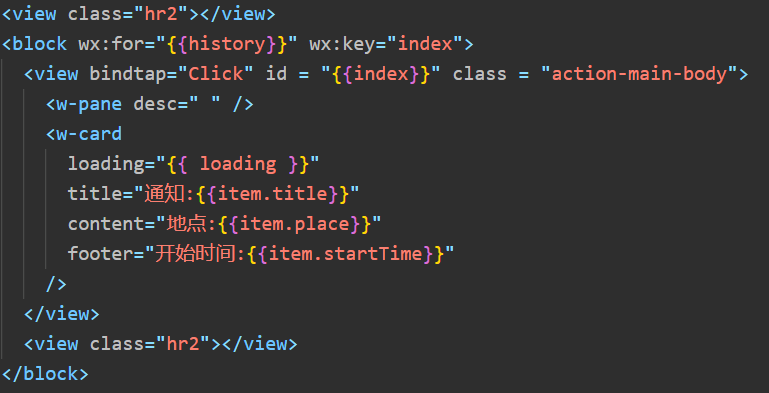
表3 历史消息查询模块变量表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 默认值 | 注释 |
| history | array |  | 用于记录历史消息 |
| item | object |  | 记录标题、正文、时间、地点、举办方 |

表4 函数记录表

|  |  |
| --- | --- |
| 函数名 | 注释 |
| Onload | 实现页面加载，同时获取数据库的数据，最多获取10条 |
| Click | 点击进入具体消息展示页面 |

actionList.wxml文件负责简要历史消息列表的展示，处理如下：

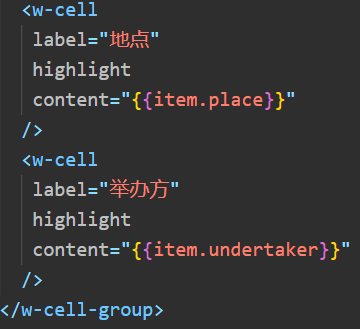
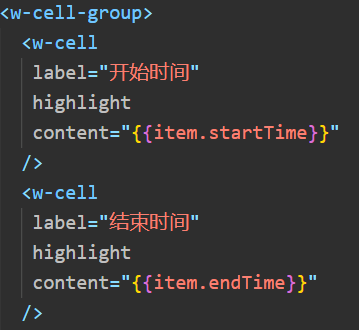


我们利用wx:for来读取我们在数据库中获得的10条最新历史消息，绑定好js文件中的Click函数，使得我们在点击某条简要信息时，可以进入具体消息展示页面。同时使用w-card组件来展示简要的消息通知。

actionDetail.wxml文件负责详细消息的展示，其处理如下：



首先我们使用w-card组件将活动的标题以及正文内容进行展示，接着我们利用w-cell组件讲其余信息进行展示，处理如下：



流程图和操作流程如下：

(1)我们点击活动模块，可以进入活动界面，我们选择点击“历史消息”模块，就可以进入历史消息的查询模块。

(2)在历史消息查询模块中，我们可以看到简要信息的展示列表。这里的简要消息主要包括活动的标题，地点和起始时间的展示。如果我们需要了解某一个活动的详细消息，点击该条活动的简要消息卡片，进入下一步。

(3)点击后进入该活动的详细消息展示页面，包括了活动的标题、正文、起始时间、结束时间、地点和举办方信息。

(4)点击小程序左上角的返回键，可以退出活动的详细信息展示界面，返回简要信息的展示列表，如果有其他想要查询的活动可以点击其他活动的简要消息卡片，否则流程结束。

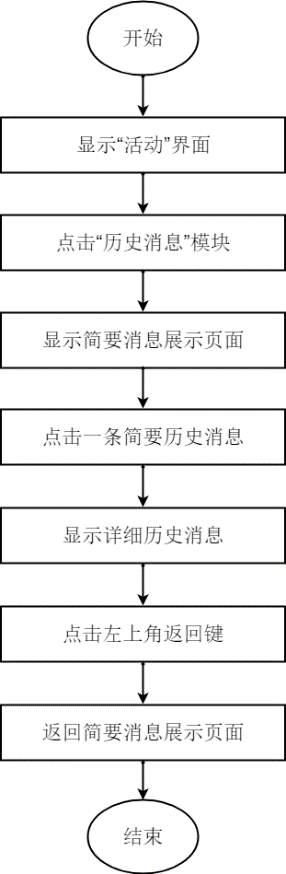


图2 历史消息查询模块流程图

5.3 最新历史消息滚动展示模块

该模块的UI设计同样是基于Wuss Weapp实现的，代码实现放在activity模块。js文件中变量和函数的设计上如下：

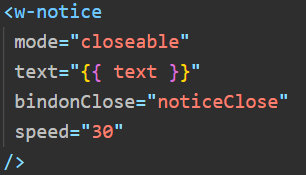
表5 最新历史消息滚动展示模块变量表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 默认值 | 注释 |
| text | string |  | 用于记录最新消息并显示 |

表6 函数记录表

|  |  |
| --- | --- |
| 函数名 | 注释 |
| Onload | 实现页面加载，同时发布最新消息的通知 |

在activity.wxml中，关于该滚动模块的处理如下：



我们将该消息滚动模块设置为可关闭模式，字体的滚动速度设置为30，text中即为读取到的最新历史消息。

6 首页模块

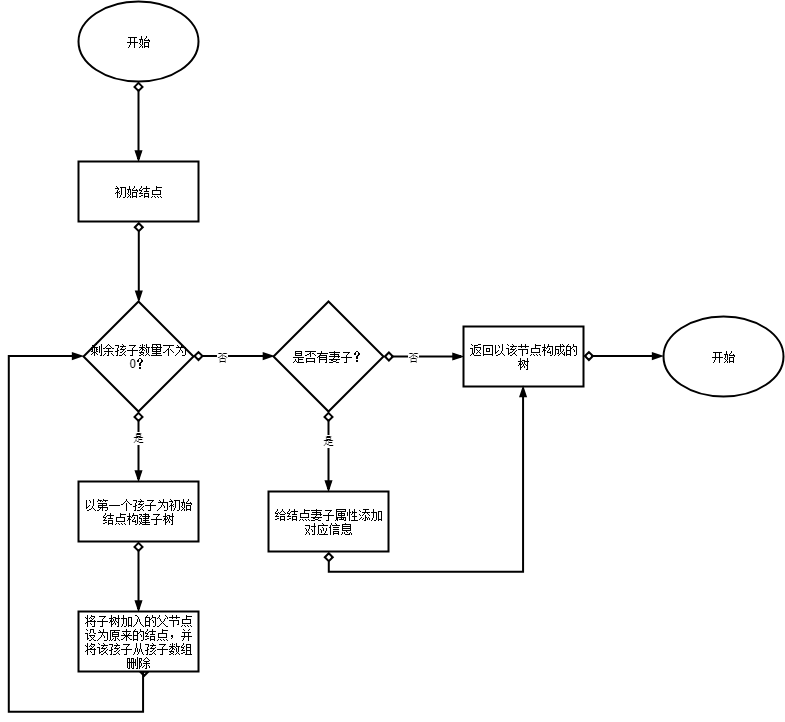
6.1族谱树模块的实现

族谱树主要使用的数据结构是数组和js的obeject，数组是为了方便查找。数组的每一个元素是一个item，其结构定义如下：

表6-1 树结构元素

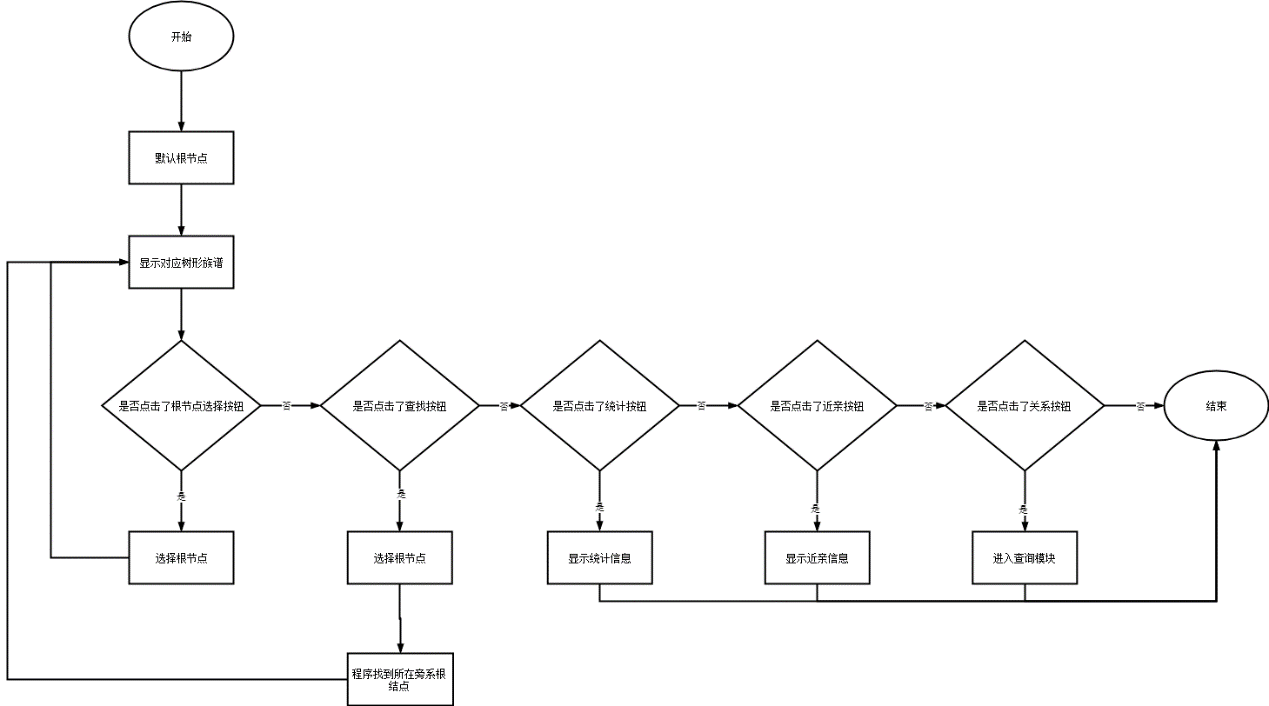
|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 默认值 | 注释 |
| ObejectId | String | 无 | 唯一识别id,对应在数据库的标识 |
| Name | String | None | 用户的名字 |
| level | Number | 无 | 辈分 |
| username | String | “” | 名字 |
| is\_wife | Boolean | False | 是否是别人的妻子 |
| children | Array | [] | 孩子的id数组 |
| child\_index | Array | [] | 孩子在数组中的下标（有利于快速定位找到孩子信息） |
| wife\_index | int | None | 妻子在数组中的下标（丈夫独有） |
| husband\_index | int | None | 丈夫在数组中的下标  （妻子独有） |
| father\_index | int | None | 父亲在数组中的下标 |

查找树首先要构建好一颗树，逻辑如下：



在代码中采用递归的思想，通过遍历一个用户的孩子，分别以孩子为根构建子树，得到一颗完整的树。树会显示在首页，树的显示是以view为基础的，集成了tree-chart组件。

用户查看树的流程如下：



(1)进入查看模块后，程序默认根结点表第一个为当前根结点，构建树状族谱显示

(2)点击左上角的根结点列表可以切换支系。

(3)点击查找按钮选择人可进入对应支系。

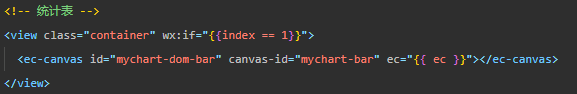
(4)点击统计按钮会生成统计信息，各年龄段男女人数。

(5) 点击上方近亲栏目会显示上下三代信息。

(6)点击关系栏目进入关系查询模块。

6.2统计模块的实现

统计信息的显示是借助echart实现的，它使用起来比较简单，在布局文件中只需要三行：

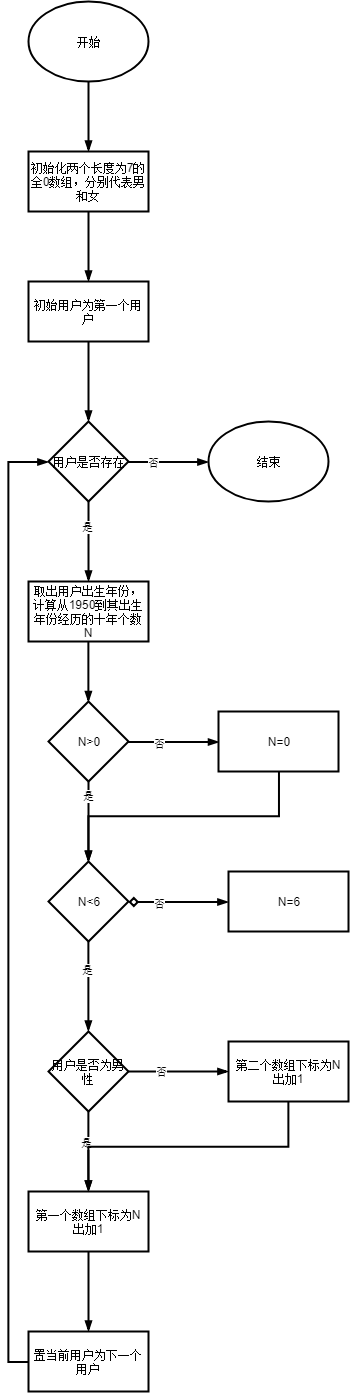


绘制的关键在于ec的传入，它是一个字典，定义如下：

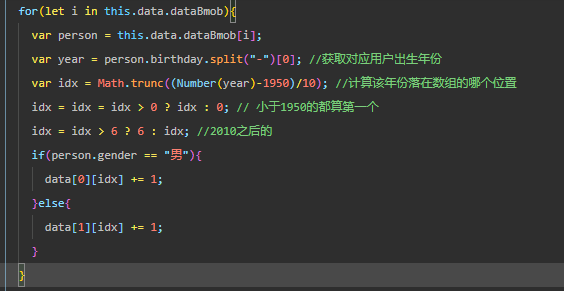
表6-2 统计信息结构元素

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 默认值 | 注释 |
| color | Array | 无 | 对应每一个柱形的颜色 |
| tooltip | Object | {} | 用户点击时弹出的提示 |
| legend | Array | [] | 图例 |
| xAxis | Array | [] | X轴刻度 |
| yAxis | Array | [] | Y轴刻度 |
| series | Array | [] | 不同组的人数数组 |

该模块的主要工作就是通过生日计算series。计算逻辑如下：



1. 初始两个包含数组表示不同性别对应年龄段的用户的人数，如下
2. 遍历用户，计算每个用户的年龄所在的数组对应位置，使数值加一 。

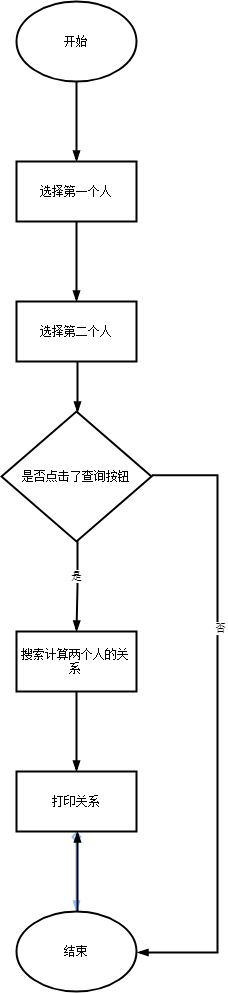


6.3关系模块的实现

因为不同地区关系称呼有所不同，我们与用户沟通之后使用了对方提供的称呼规则，如下：

1. 四代以内，男性直系称呼依次为“爸爸”，“爷爷”，“公太”，女性直系称呼依次为“妈妈”，“奶奶”，“太奶奶”。
2. 与自己同辈的，如果父亲相同，则称长者为“哥哥”，“姐姐”，年纪比自己小的为“弟弟”，“妹妹”。否则加上“堂”字。
3. 比自己长一辈的，如果跟自己父亲是亲生兄弟，则称呼比父亲大的男性为“伯伯”，其妻子称为“伯母”，小的为“叔叔”，其妻子称为“婶婶”如果跟自己父亲不是亲生兄弟，在前面加上“堂”字即可，比如“堂叔叔”，其妻子称为“堂婶婶”。其余女性，如果跟父亲是亲生兄妹，无论大小都称为“姑姑”，否则为“堂姑姑”。
4. 比自己长两辈的，如果跟自己爷爷是亲生兄弟，则称呼比爷爷大的男性为“伯公”，其妻子称为“伯婆”，小的为“叔公”，其妻子称为“叔婆”如果跟自己父亲不是亲生兄弟，在前面加上“堂”字即可，比如“堂叔公”，其妻子称为“堂叔婆”。其余女性，如果跟爷爷是亲生兄妹，无论大小都称为“姑婆”，否则为“堂姑婆”。
5. 比自己大三辈的，如果跟自己公太是亲生兄弟，则称呼比公太大的男性为“伯公太”，其妻子称为“伯婆太”，小的为“叔公太”，其妻子称为“叔婆太”如果跟自己父亲不是亲生兄弟，在前面加上“堂”字即可，比如“堂叔公太”，其妻子称为“堂叔婆太”。其余女性，如果跟公太是亲生兄妹，无论大小都称为“姑婆太”，否则为“堂姑婆太”。
6. 比自己大四辈以上，视其大的辈分与三辈之差在前面加“曾”，如“曾曾爷爷”

关系的计算以上述为依据，同时关系模块的布局为两个输入用户的按钮以及一个查询按钮，如下:



(1)点击左按钮进入搜索界面，通过姓名跟生日选择第一个人。

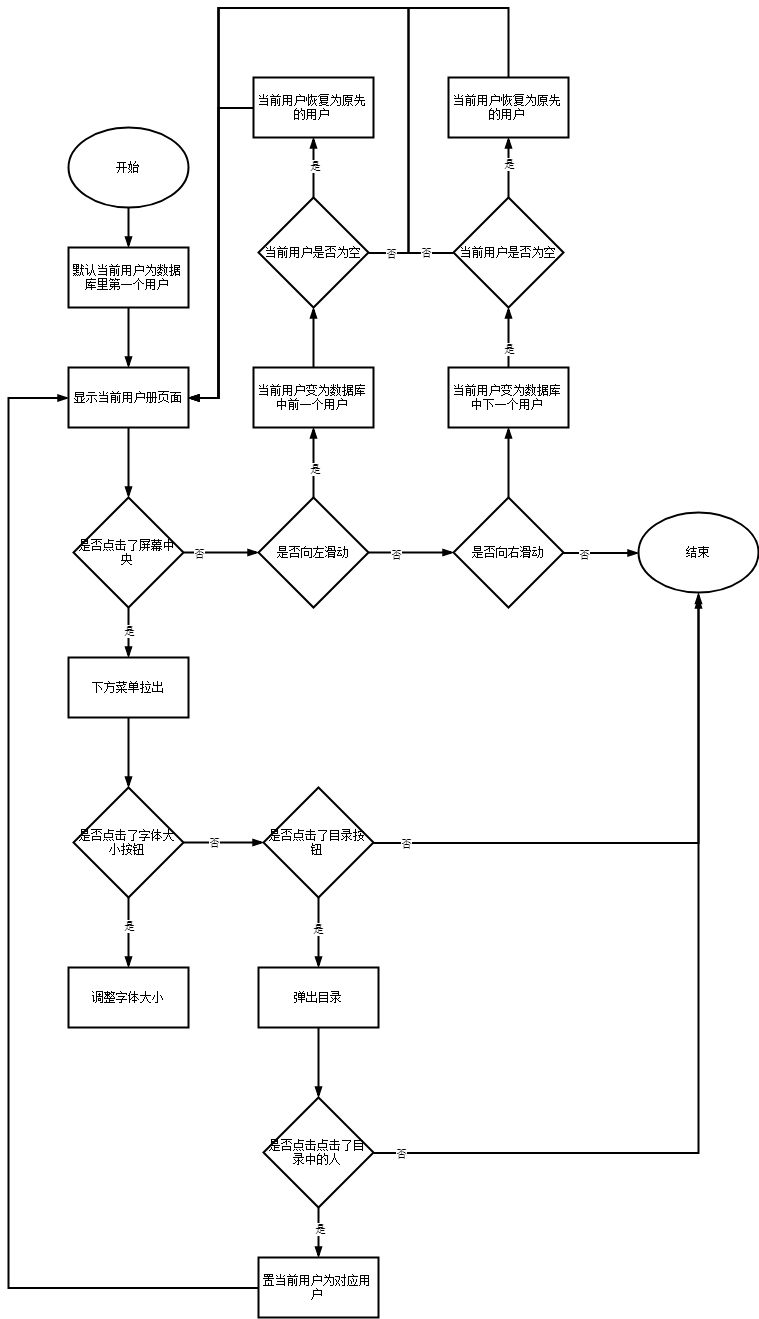
(2)点击右按钮进入搜索界面，通过姓名跟生日选择第二个人。

(3)如果点击了查询按钮，程序搜索计算两人关系。关系呈现方式是年纪大的相对于小的关系，比如某某是某某的爸爸等等，如果没有选择够两个人程序是无法计算的，会弹出提醒用户进行选择的提示。

(4)打印两个人的关系在频幕上。

7成册模块

成册模块相对于其他的模块没有太多的页面切换，这个模块是通过仿书籍的方式让用户去“翻阅”族谱信息，用户进入该模块后，流程如下：

(1)显示第一个用户页面。

(2)如果点击中间屏幕则弹出菜单转(3)，否则转(5)。

(3)如果点击了字体，可调节字体大小。

(4)如果点击了目录按钮，弹出目录选择用户显示对应页面。

(5)往左滑动则显示当前用户的上一位用户，如果是第一位则不做变化。

(6)往左滑动则显示当前用户的下一位用户，如果是最后一位则不做变化。