**移动互联网导论大作业----西电创新俱乐部招新小程序**

**21009200038 江昱峰 21009201055 李旭阳 21009200155 刘冬颍**

**团队成员分工：**

江昱峰：小程序后端&发布、设计文档编写

李旭阳：小程序后端&运维、答辩

刘冬颍：小程序前端、PPT

**程序功能：**

我们小组实现的是西电创新俱乐部招新的小程序，里面的功能包括开篇的上滑进入页面操作，填写自己的学号姓名，手机号码等基本信息，填写成功之后跳转到西电创新俱乐部的相关推文推荐内容。

普通用户（报名、了解的同学）：编辑个人介绍，查看俱乐部所有部门、小组介绍和面试要求，报名（面试前）&接收录取结果通知（面试后）。实现效果如下：



图 1：个人介绍编辑功能



图 2：部门小组介绍功能



图 3：报名和接收录取结果通知功能

管理员（组长、部长、主席等面试负责人）：除了普通用户的全部功能外，还能查看小程序每日浏览量、注册用户数、各部门小组面试报名人数等数据的可视化结果，并结合面试结果选择录取或拒绝报名同学。实现效果如下：

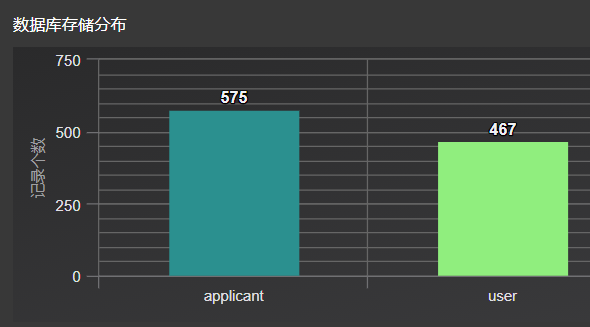


图 4：后台数据可视化

**设计思想：**

**其中微信小程序要得以实现，就需要必须的几个部分：**

总体来说，就是根据不同的程序功能，相应地用函数式编程的方法，设计函数方法来实现。具体设计思想分为后端和前端两块：

后端：用户个人信息，后台浏览数据，报名信息；

前端：各部门小组招新海报，面试报名和通知界面。

微信小程序通常需要提供以下类型的文件才能完整运行：

1. 小程序主体部分：

app.js： 小程序的全局脚本文件，定义小程序的全局配置和生命周期函数。

app.json： 小程序的全局配置文件，定义页面路径、窗口样式、网络超时时间等全局配置信息。

app.wxss： 小程序的全局样式文件，定义小程序的全局样式。

1. 页面相关文件：

wxml： 小程序的页面结构文件，定义页面的结构。

wxss： 小程序的页面样式文件，定义页面的样式。

js： 小程序的页面逻辑文件，定义页面的行为和业务逻辑。

json： 小程序的页面配置文件，定义页面的配置信息，如导航栏样式、是否允许下拉刷新等。

1. 其他资源文件：

图片文件（.png、.jpg等）： 用于小程序中的图片资源。

字体文件（.ttf、.woff等）： 用于小程序中的字体资源。

音频文件（.mp3、.aac等）： 用于小程序中的音频资源。

1. 云函数：

云函数文件夹： 如果使用了云开发功能，需要提供包含云函数代码的文件夹，其中至少包含 index.js、package.json 等云函数所需的文件。

1. 其他配置文件：

project.config.json： 小程序项目的配置文件，包含了项目的一些配置信息。

.gitignore： 如果使用 Git 进行版本控制，可以提供 Git 忽略文件，避免将不必要的文件提交到版本库中。

以上是基本的小程序文件类型，具体项目可能还会根据需求包含其他类型的文件。

所以我们同样采用了以上代码安排来实现这些功能，app.js app.json app.wxss代码共同实现了小程序的整体部分，而页面分布采用了wxml，js，json后缀的三部分文件根据不同功能写入了如下功能：

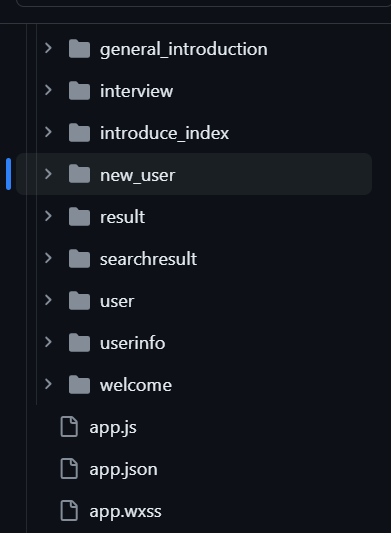


图 5：小程序的方法函数目录

Gengral\_introduction实现了对西电创新俱乐部的大致介绍部分

Interview实现了对西电创新俱乐部内部人员对申请者的同意或拒绝部分

Introduce\_index实现了对西电创新俱乐部的小组信息部分

New\_user实现了提交申请之后通过成为新成员的部分

Result实现了最终结果的部分

Srearchresult实现了提交申请之后是否得到录取通知的部分

User实现了西电创新俱乐部的用户信息数据库

Userinfo实现了申请的用户的通信链接的部分

Welcome实现了刚开始打开小程序的欢迎界面

**其中在Introduce\_index.js代码部分中：**

1. **页面初始化**：在 onLoad 生命周期函数中，通过调用云函数 getGroups 获取全部小组的信息，并通过调用云函数 getGroupTypes 获取小组分类信息。这些信息将被用于页面的展示和操作。
2. **小组展示和选择**：其中页面中展示了小组的信息，包括小组的名称、封面图等。用户可以点击小组进行选择操作，选择的小组信息将会被保存在 selectGroupId 数组中。并且最后小组选择后，页面中相应的小组封面图会发生变化。
3. **申请状态展示**：页面中展示了申请状态的信息，包括申请状态的小红点和待处理、已通过、未通过的申请数量。
4. **点击小组查看详情**：用户可以点击小组，查看小组的详细介绍。跳转页面时，会传递小组的相关信息，如介绍图、背景图等。
5. **消息功能**：用户可以点击消息图标，弹出消息弹框。通过 showModal 和 hideModal 函数，实现消息弹框的显示和隐藏。
6. **处理申请结果展示**：通过调用云函数 getCurApplicants 获取当前用户的申请结果信息，包括已通过、未通过、待处理的申请信息，并在页面中展示。

**Welcome代码部分中**:

1. **滑动手势操作：**通过 **touchStart**、**touchMove**、**touchEnd** 函数实现了滑动手势的监听，可以识别用户的上滑操作。
2. **动画效果：**利用小程序的动画系统，通过 **wx.createAnimation** 创建动画实例，实现了头像的上下浮动效果。动画的实现使用了 **setInterval** 定时器，使得头像在上下浮动。
3. **用户登录处理：**在页面加载时，通过调用云函数 **fakeCode** 获取用户的伪登录码，并保存在全局变量 **fakeCode** 中。当用户进行上滑操作时，通过 **login** 函数调用云函数 **login** 进行用户登录的处理。如果用户未注册（返回 code 为 1002），则调用 **navigateToNewUser** 函数跳转到新用户注册页面。如果用户已注册（返回其他 code），则调用 **navigateToUserPage** 函数跳转到用户信息展示页面。
4. **动态更新头像：**在页面展示时，通过 **onShow** 函数动态更新头像路径，确保用户头像的及时展示。
5. **定时动画效果：**使用 **setInterval** 定时器，每隔一定时间切换一次动画效果，实现头像的上下浮动。
6. **用户信息的全局保存：**在登录成功后，通过 **getApp().globalData.userInfo** 将用户信息保存在全局变量中，以便其他页面可以共享和使用。
7. **页面跳转：**使用 **wx.reLaunch** 进行页面的跳转，确保用户返回时重新加载页面。

**Userinfo代码部分：**

1. **加载用户信息：**通过调用云函数 `getUserInfo` 获取指定用户的信息，并在页面加载时显示用户的基本信息，如性别、昵称等。

2.**显示用户评论信息**：通过页面的 `onLoad` 函数获取传递过来的参数，包括用户的ID和评论信息，然后通过调用云函数获取用户的信息，并显示在页面上。

3.**修改用户评论**：提供了一个文本域，允许用户修改对该用户的评论信息。

4. **保存修改后的评论**：通过点击按钮（可能是提交按钮）触发 `bindFormSubmit` 函数，将修改后的评论信息通过云函数 `saveComment` 保存到数据库中。

5. **返回上一页**：在保存成功后，通过 `wx.navigateBack` 返回到上一页（即用户列表页或其他上一页）。

**User代码部分：**

1. **页面数据初始化：**在页面加载时，通过 onLoad 生命周期函数初始化页面数据，包括用户信息、是否管理员、头像等。
2. **查看个人信息：**提供一个按钮（goToUserInfoFake函数触发），点击后跳转到用户信息页面，其中用户信息的数据来自全局数据（getApp().globalData.userInfo）。
3. **展示面试消息：**利用 wx.cloud.callFunction 调用云函数获取面试消息，并在页面加载完成后显示在页面上**。**
4. **导航跳转：**提供多个按钮，点击后可以导航到其他页面，如展示俱乐部介绍页面、进行面试等。
5. **编辑用户信息：**提供编辑用户信息的功能，点击后跳转到编辑用户信息页面。
6. **显示模态框：**提供一个按钮（showMessage函数触发），点击后显示模态框，展示用户信息中的一些消息。
7. **修改用户头衔：**提供一个按钮（editTitle函数触发），点击后弹出模态框，用户可以在模态框中输入新的头衔，然后调用云函数进行修改。

**Sreachresult代码部分：**

1. **页面加载（onLoad）：**通过接收传递的参数 options.stuinfo，解析为学生信息，并在页面加载时进行展示。
2. **学生信息展示：**在页面加载时，展示学生的相关信息，包括学生的状态（通过、未通过、处理中等）**。**
3. **用户信息跳转：**点击学生信息，可以跳转到相应用户的详细信息页面。
4. **同意/拒绝操作：**用户可以通过点击同意或拒绝按钮，弹出确认框，确认后将学生的申请状态更新为通过或拒绝。
5. **状态更新后刷新页面：**操作成功后，页面会重新加载并刷新显示最新的学生信息和状态。
6. **异步操作的提示：**在异步操作（如调用云函数）进行时，显示加载中提示，操作完成后显示成功提示。

**Result代码部分：**

1. **数据展示：**通过 onLoad 生命周期函数获取传递过来的参数 options，包括 group、clubQQ 和 groupQQ。将获取到的数据存储在页面的数据 data 中。
2. **文本复制：**提供了 copyText 函数，当用户点击相关文本区域时，将相应的文本内容复制到剪贴板。利用 wx.setClipboardData 和 wx.getClipboardData 接口实现文本的复制操作。显示复制成功的提示信息，通过 wx.showToast 实现。
3. **页面加载时的初始化：**在页面加载时，通过 onLoad 设置页面的初始数据，包括 groupName、clubQQ 和 groupQQ。
4. **分享功能：**通过 onShareAppMessage 监听用户点击右上角分享按钮的事件，可以自定义分享的内容。
5. **生命周期函数：**页面提供了一些生命周期函数，例如 onReady、onShow、onHide、onUnload，用于在不同的生命周期阶段执行相应的操作。

**New\_user代码部分：**

1. **选择头像：**通过 onChooseAvatar 函数，响应用户选择头像的操作。该函数获取用户选择的头像地址，并将其更新到页面的 avatarUrl 数据中。
2. **提交虚拟信息：**通过 submitFake 函数，将用户选择的虚拟昵称和之前选择的头像信息保存到全局数据中。然后，通过 wx.navigateTo 跳转到用户信息展示页面。
3. **加载页面时的初始化：**在 onLoad 生命周期函数中，获取全局数据，包括用户的 OpenID 和虚假的验证码（fakeCode），并初始化页面的一些数据。
4. **表单提交：**通过 bindFormSubmit 函数，监听表单提交事件，获取用户输入的表单数据，并进行一些基本的验证，如检查是否填写了所有字段。然后，将用户头像上传到云端，并将用户信息保存到云数据库中。
5. **表单验证：**checkFormData 函数用于验证用户输入的表单数据是否完整，包括姓名、性别、年级、专业、电话、邮箱、个人介绍等字段。

**Interview代码部分：**

1. **申请信息展示：**通过调用云函数 getApplicants 获取申请者的信息，包括姓名、状态等。
2. **搜索功能：**用户可以通过输入关键词，触发 search 函数调用云函数 searchApplicants 进行搜索，返回匹配的申请者信息。
3. **搜索结果展示：**当搜索成功时，通过 wx.navigateTo 跳转到 searchresult 页面，展示搜索到的申请者信息。
4. **审批通过/拒绝：**管理员可以通过页面上的按钮对申请者进行审批，通过 Agree 和 disAgree 函数，调用云函数 saveResult 完成审批操作，同时更新页面显示。
5. **隐藏/显示状态：**通过 is\_hidden 数组控制每个申请者的信息在页面上的显示状态，实现审批通过后隐藏该项。
6. **页面生命周期处理：**利用页面的生命周期函数，如 onLoad、onShow 等，在页面加载、显示时调用云函数获取数据，并在需要时刷新页面。
7. **异步操作处理：**使用异步操作，例如云函数的调用，通过 wx.cloud.callFunction 异步获取数据，避免阻塞页面加载。

**关键技术点：**

从编程语言方面，关键技术主要在于JavaScript、CSS、HTML三件套的配合使用；从程序功能方面，主要在于包含同学报名和接收录取结果通知、面试负责人接收/拒绝两块功能的报名系统这一核心功能。

**Introduce\_index.js代码部分中：**

1. **小程序生命周期函数：**使用了小程序的生命周期函数，如 **onLoad**、**onShow** 等。这些函数在页面加载、显示等不同阶段会被触发，用于执行相应的操作。
2. **云开发（Cloud Development）：**利用了微信小程序的云开发能力，包括调用云函数和访问云数据库。通过 **wx.cloud.callFunction** 调用云函数，实现与后端的交互。调用了 **getGroups**、**getGroupTypes**、**getCurApplicants**、**fakeCode** 等云函数来获取相关数据。
3. **采用了页面跳转和参数传递：**利用 **wx.navigateTo** 实现页面之间的跳转，通过 **url** 参数传递需要的数据。在页面的 **onLoad** 函数中通过 **options** 参数获取传递过来的数据。
4. **数据绑定和更新：**利用了小程序的数据绑定机制，通过 **setData** 函数更新页面数据。数据的双向绑定使得页面能够及时响应数据的变化，实现动态展示。
5. **事件处理和交互：**使用了小程序的事件处理机制，如 **bindTap**、**bindInput** 等，实现用户的交互操作。例如，点击小组进行选择、点击消息图标弹出消息弹框等。
6. **异步操作和 Loading 提示：**利用了异步操作，在调用云函数或进行数据获取时，使用 **wx.showLoading** 和 **wx.hideLoading** 提示用户等待。

**Welcome代码部分中：**

1. **滑动手势监听：**使用 touchStart、touchMove、touchEnd 函数来监听用户的滑动手势，通过计算触摸点的坐标变化来判断用户的上滑操作。
2. **动画效果的实现：**使用小程序提供 wx.createAnimation 创建动画实例，通过设置动画的属性（如 translateY）实现头像的上下浮动效果。
3. **云函数调用：**通过 wx.cloud.callFunction 调用云函数，实现获取伪登录码和用户登录的操作。这涉及到小程序与云开发的集成和通信。
4. **定时器的使用：**使用 setInterval 定时器来触发动画效果的切换，定时更新头像的上下浮动。
5. **用户信息的全局保存：**利用小程序的全局变量 getApp().globalData 保存用户信息，以便在其他页面中获取和使用用户信息。
6. **页面跳转：**使用 wx.reLaunch 进行页面的跳转，重新加载目标页面，确保用户返回时重新执行页面加载操作。
7. **条件判断和处理：**利用条件语句和逻辑判断，根据云函数返回的状态码实现不同的处理逻辑，例如判断用户是否已注册。
8. **小程序生命周期函数的使用：**利用小程序的生命周期函数，如 onLoad、onShow 等，在不同的生命周期阶段执行不同的操作，确保页面加载和展示时的一致性。

**Userinfo代码部分：**

1. **调用云函数：**通过 **wx.cloud.callFunction** 方法调用云函数，实现对云端数据库的读取和操作。
2. **异步数据加载：**通过使用 **wx.showLoading** 和 **wx.hideLoading** 方法，在数据加载过程中提供用户友好的加载提示，确保数据加载完成后再显示页面。
3. **动态页面数据绑定：**利用页面的 **onLoad** 生命周期函数，根据用户ID加载对应的用户信息，并通过 **setData** 更新页面数据，实现动态数据的绑定与展示。
4. **表单提交与数据保存：**利用表单组件，通过 **bindFormSubmit** 方法监听表单提交事件，获取用户输入的评论信息，并通过云函数将评论信息保存到数据库中。
5. **页面间参数传递：**通过页面路径中的参数（如 **options.userId** 和 **options.comment**）获取其他页面传递过来的数据。

**User代码部分：**

1. **小程序页面生命周期使用：**使用小程序页面的生命周期函数，如 onLoad、onShow 和 onReady，来在不同阶段执行相应的逻辑。
2. **全局数据的使用：**利用 getApp().globalData 来获取和设置全局数据，使得不同页面之间可以共享数据。
3. **调用云函数：**使用 wx.cloud.callFunction 调用云函数，从云端获取数据或执行相应的操作。
4. **页面跳转：**使用 wx.navigateTo 和 wx.reLaunch 等方法实现页面之间的跳转和重新加载。
5. **动画效果：**利用小程序动画 API（wx.createAnimation）实现用户界面上的动画效果，如图标的平移。
6. **模态框的使用：**使用 wx.showModal 实现模态框的显示，用户可以在模态框中输入信息或进行选择。
7. **云函数中数据库操作：**在云函数中进行数据库的读取和更新操作，包括获取面试消息和更新用户头衔等。
8. **用户交互事件处理：**对用户的触摸事件进行处理，包括 touchStart、touchMove 和 touchEnd 等，以实现滑动或其他手势的交互操作。

**Sreachresult代码部分：**

1. **参数传递：**使用 onLoad 生命周期函数接收页面跳转时传递的参数，其中 options.stuinfo 包含学生的信息。
2. **学生信息展示：**利用页面数据绑定，将学生信息渲染到页面上。
3. **用户信息跳转：**利用小程序的页面导航功能，通过 wx.navigateTo 实现页面跳转，并通过 URL 参数传递用户相关信息。
4. **异步数据加载：**使用异步操作，例如调用云函数 wx.cloud.callFunction 获取数据，并在加载完成后更新页面。
5. **操作确认框：**利用 wx.showModal 实现操作的确认框，用户可以在确认框中选择同意或拒绝操作。
6. **状态更新和页面刷新：**根据用户的同意或拒绝操作，调用云函数更新学生的状态，并通过刷新页面展示最新的学生信息。
7. **加载提示和操作成功提示：**使用 wx.showLoading 在异步操作进行时显示加载中提示，操作完成后使用 wx.showToast 显示成功提示。
8. **数组操作：**对 is\_hidden 数组进行操作，根据学生状态更新该数组，影响页面上同意、拒绝按钮的显示状态。

**Result代码部分：**

1. **页面生命周期：**使用小程序提供的生命周期函数，如 **onLoad**，**onReady**，**onShow**，**onHide**，**onUnload** 等，来执行相应的初始化和清理操作。
2. **数据传递：**通过 **onLoad** 生命周期函数获取从其他页面传递过来的参数，即 **options** 对象中的数据。
3. **数据展示：**利用页面的数据绑定机制，将获取到的数据展示在页面上，例如通过 **{{ groupName }}** 的方式将 **groupName** 的值渲染到页面中。
4. **剪贴板操作：**通过调用小程序 API 中的 **wx.setClipboardData** 和 **wx.getClipboardData** 接口，实现文本的复制到剪贴板和从剪贴板读取的功能。
5. **分享功能：**通过 **onShareAppMessage** 监听用户点击分享按钮的事件，返回自定义的分享内容，实现页面的分享功能。
6. **页面跳转：**页面中可能包含跳转到其他页面的功能，例如通过 **wx.navigateTo** 实现跳转到其他页面的操作。
7. **事件处理：**页面中的点击事件，例如点击文本复制的操作，通过绑定相应的事件处理函数实现。

**New\_user代码部分：**

1. **头像选择与显示：**使用 onChooseAvatar 函数响应用户选择头像的操作，通过 setData 更新页面的头像显示。
2. **虚拟信息提交：**通过 submitFake 函数保存用户选择的虚拟昵称和头像信息到全局数据中，然后跳转到用户信息展示页面。
3. **页面初始化：**在 onLoad 生命周期函数中，获取全局数据，包括用户的 OpenID 和虚假的验证码（fakeCode），用于页面初始化**。**
4. **表单提交与验证：**使用 bindFormSubmit 函数监听表单提交事件，获取用户输入的表单数据，并通过 checkFormData 函数验证表单数据是否完整。头像通过云上传，然后将用户信息保存到云数据库中。
5. **云存储：**使用小程序云开发的 wx.cloud.uploadFile 方法将用户选择的头像上传至云端，并获得文件 ID。这个文件 ID 用于保存到用户信息中，实现头像的云存储。
6. **全局数据的使用：**通过 getApp().globalData 获取全局数据，包括用户信息、OpenID、虚假验证码等。这些全局数据用于不同页面之间的数据传递和共享。
7. **页面跳转：**使用 wx.navigateTo 和 wx.reLaunch 进行页面之间的跳转，实现用户选择虚拟信息后的流程控制。

**Interview代码部分：**

1. **云函数的使用：**通过 wx.cloud.callFunction 调用云函数，实现在云端执行的逻辑，例如获取申请者信息、保存审批结果等。
2. **数据的获取与展示：**利用小程序页面的生命周期函数（如 onLoad、onShow）调用云函数获取数据，然后通过数据绑定更新页面展示。
3. **搜索功能的实现：**利用输入框的输入事件，触发搜索函数，通过 wx.cloud.callFunction 调用云函数进行搜索，返回匹配的结果。
4. **页面跳转与参数传递：**使用 wx.navigateTo 实现页面跳转，通过参数将数据传递到下一个页面，例如搜索结果传递到 searchresult 页面。
5. **审批操作与状态更新：**通过调用云函数实现对申请者的审批操作，审批结果保存在云数据库中，并通过更新页面数据实时反映审批状态。
6. **异步操作处理：**利用异步操作避免阻塞页面，例如在获取数据、调用云函数时使用 await 或者 Promise 确保异步操作的顺序执行。
7. **数组状态管理：**使用数组（例如 is\_hidden）来管理申请者信息的显示状态，根据审批结果实时更新数组，实现信息的显示与隐藏。
8. **事件处理：**通过事件处理函数，如 Agree 和 disAgree 处理用户的点击操作，触发相应的审批逻辑。

**结果展示：**

本小程序已经经过微信审核并分布，且已用于本届社团招新，此处提供该小程序的名称和二维码，供老师搜索：



图 6：微信小程序二维码