

第十八组

第十八组答辩

组长：李世娇

组员：廖谊、敬嘉月、石伟琪



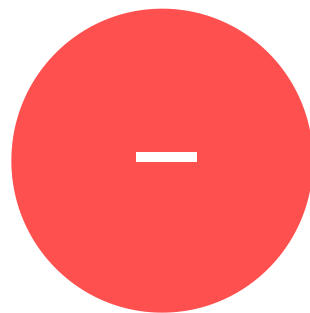
目录

1.项目概述

2.成员分工

3.设计过程

4.总结收获



项目概述



携程旅行APP

携程产品定位产品概述：在线票务服务公司（OTA），创立于1999年，总部设在上海。具有完整产业生态系统，直接为酒店、航空、铁路、旅游景区管理、保险等公司提供代理销售和广告业务，为消费者提供涵盖出行、住宿、保险、旅游等所有旅行服务的一站式服务平台。

Solgan：携程在手，说走就走

1.项目概述

uniAPP-携程旅行

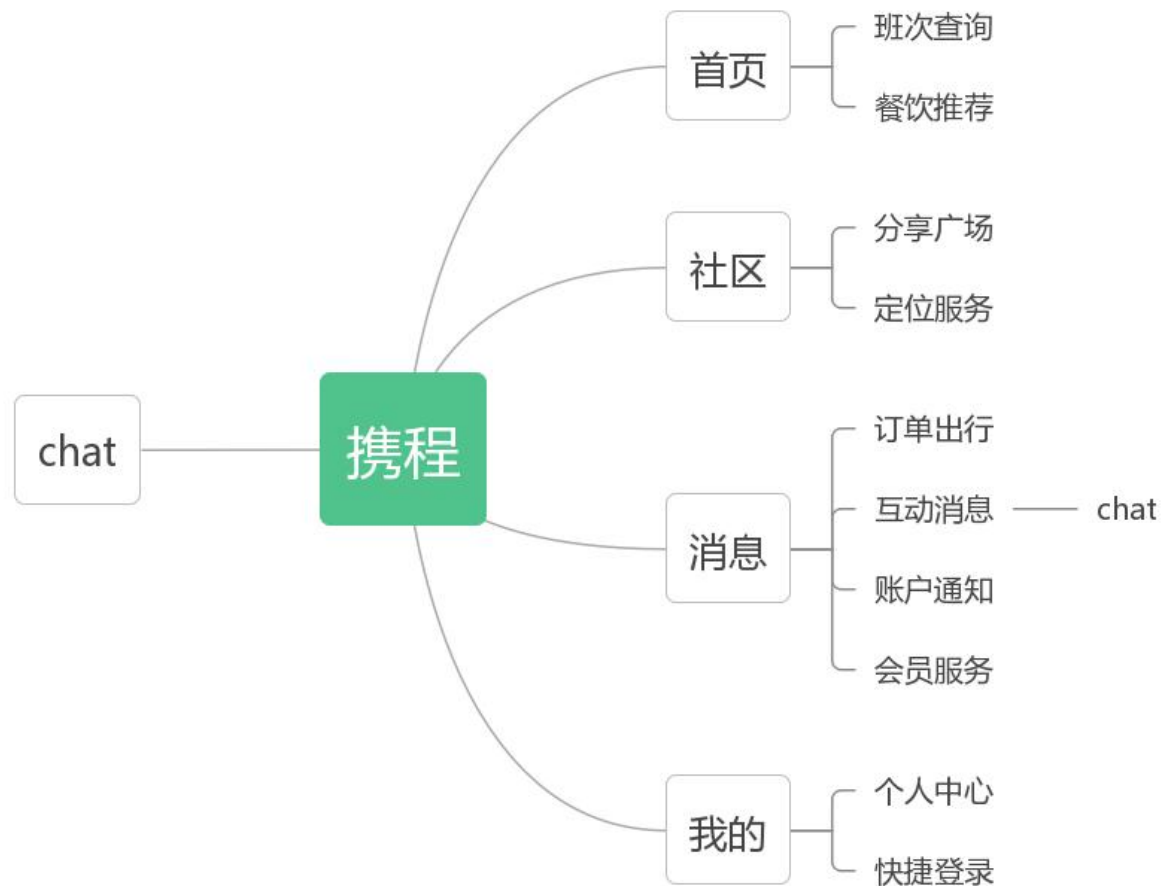
pages-首页

pages-社区

pages-消息

pages-我的

gptAPI-chat



界面展示



首页



社区



消息



我的

1.项目概述

uniAPP-携程旅行

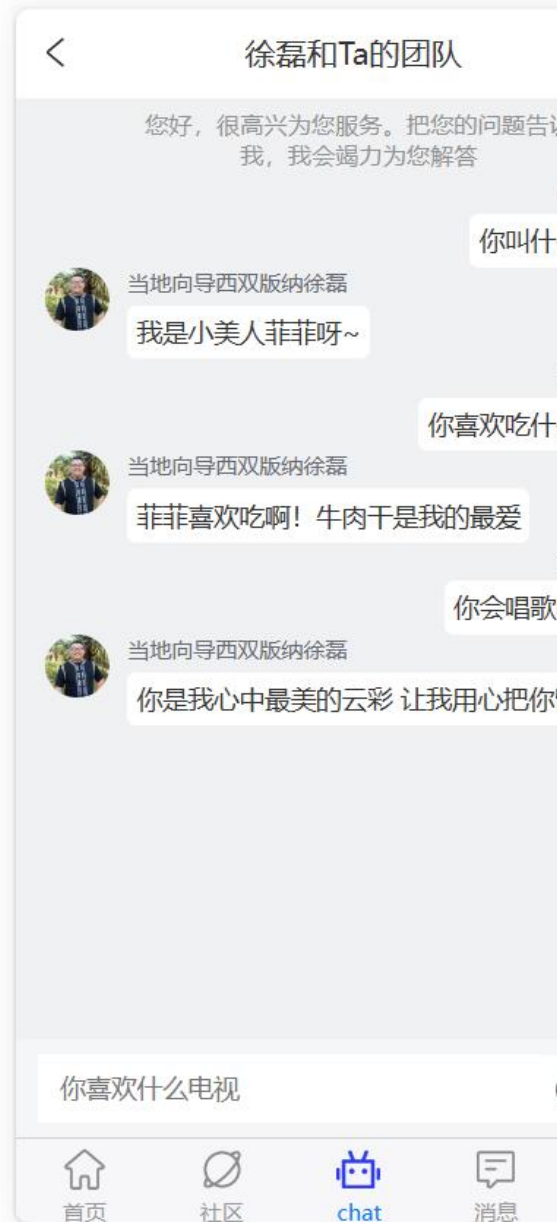
pages-首页

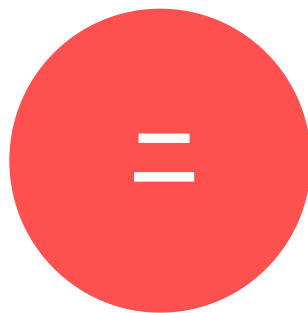
pages-社区

pages-消息

pages-我的

gptAPI-chat





成员分工

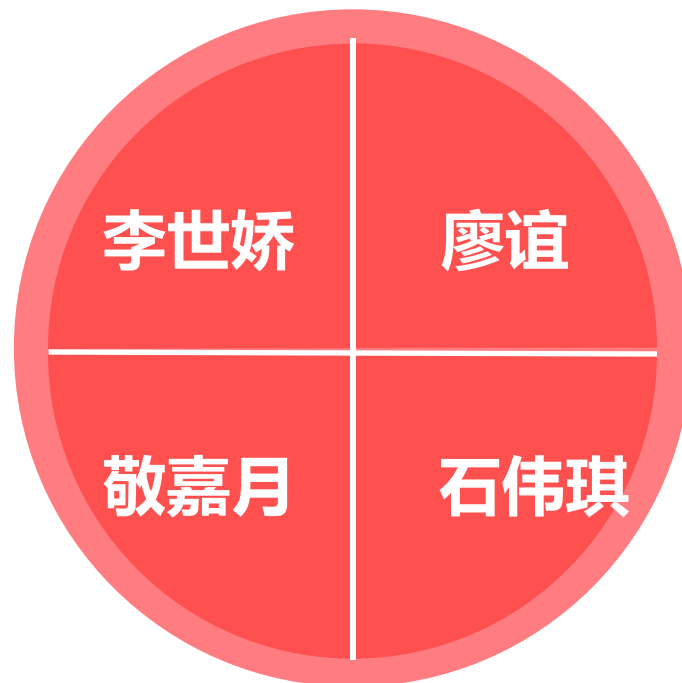
成员分工

消息+Chat

界面设计：消息页面及其子页面
交互设计：对话功能、跳转功能、可滚动视图

社区

界面设计：社区页面及其子页面
交互设计：跳转功能、可滚动视图、定位菜单栏



首页

界面设计：首页及其子页面
交互设计：跳转功能、轮播图、弹窗提示、显示消息提示框

我的

界面设计：个人页面及其子页面
交互设计：跳转功能、登录功能、弹窗提示、显示消息提示框



设计过程

设计过程

界面设计：消息页面及其子页面

交互设计：对话功能、跳转功能、可滚动视图

消息+Chat



首页



社区



消息

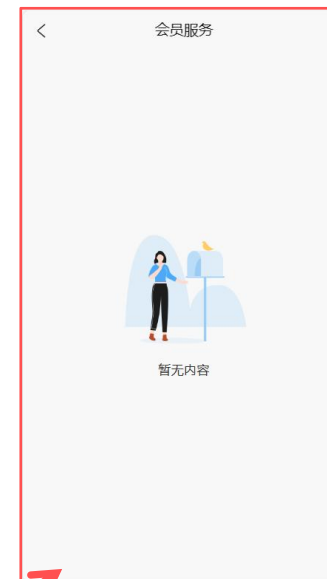
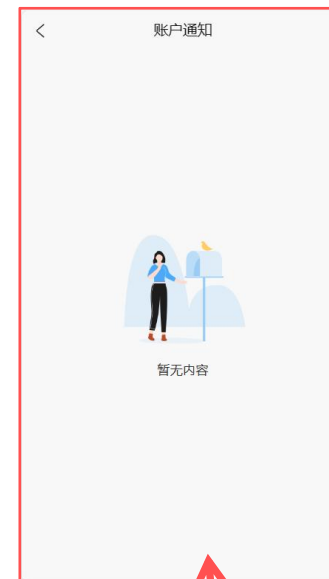


我的

设计过程

主要页面

消息+Chat



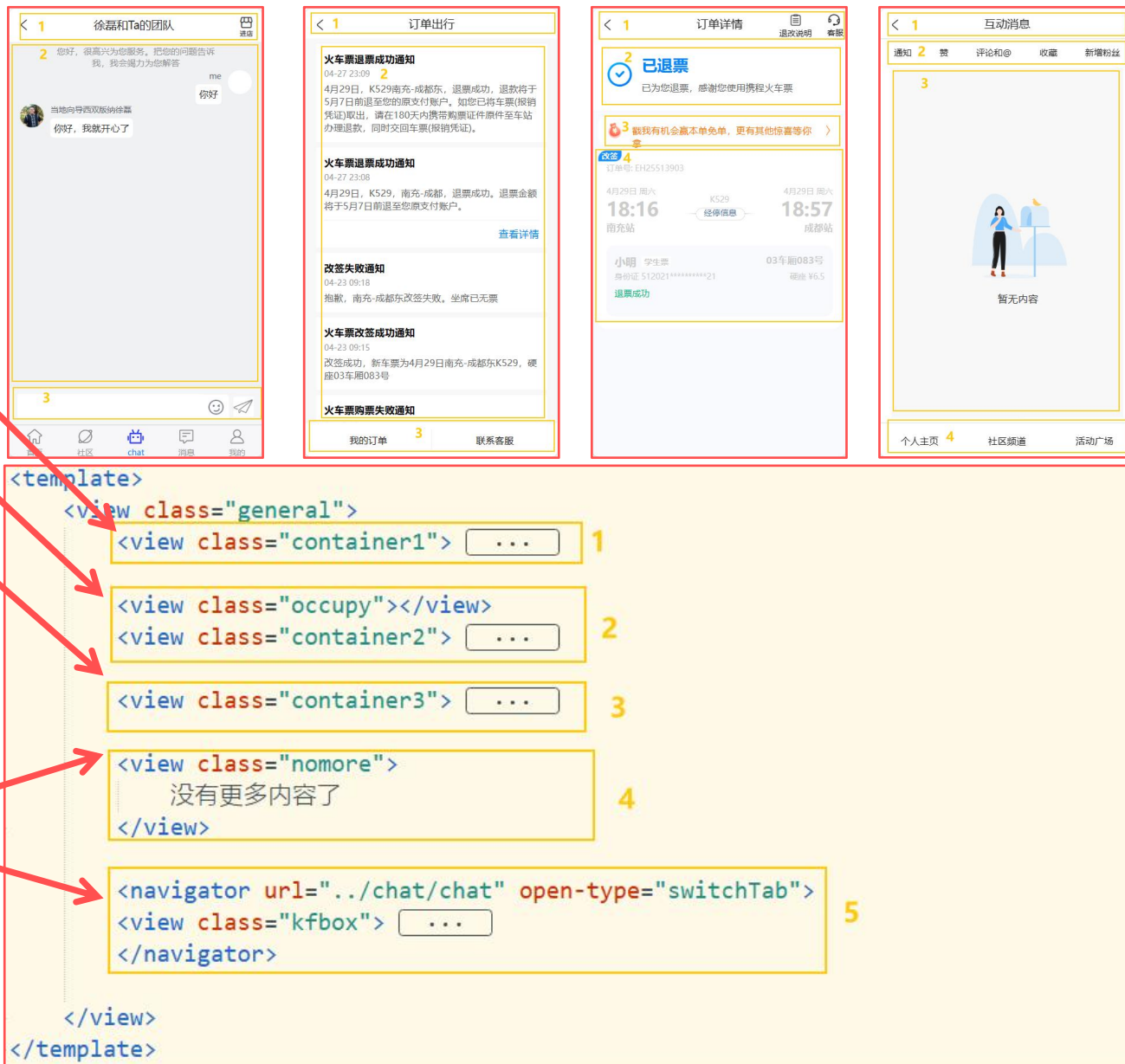
设计过程

划分布局

消息+Chat



消息





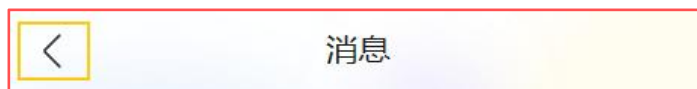
v-for="(item,index) in
arr 渲染追加单元格



消息



v-if=" item.isnt" 条件渲染,
引号内为真值时渲染该条件



open-type,默认navigate,用switchTab



globalStyle:用于设置应用的状态栏、
导航条、标题、窗口背景色等。
navigationStyle 默认值default
custom即取消默认的原生导航栏



justify-content: space-between
两端对齐, 空白均匀地填充到 flex
成员项之间



客服盒子使用fixed, 始终相对于模
拟器定位, 不随滚动条移动



列表渲染

消息



简洁代码



v-for v-if v-else

v-for=" (item,index) in arr" 渲染追加单元格

v-if=" boolean" 为true条件渲染 (false不渲染)

v-else 否则不渲染

v-on @click

事件绑定 method中写方法

v-bind : {}

数据绑定data中写数据

```
<view class="chatson" v-for="(item,index) in chatarr">
  <image :src="item.imgsrc" mode="" class="hd"></image>
  <view class="ctreddot" v-if="item.isnew"></view>
  <view class="chattext">
    <view class="info">
      <view class="time">{{item.time}}</view>
      <view class="name">{{item.name}}</view>
      <view class="note">
        <view class="nttext" v-if="item.isnew">发现新内容</view>
        <image src="/static/message/8note.png" mode="" class="ntimg" v-if="item.isnew" :data-id="item.id" @click="click"></image>
      </view>
      <view class="content">{{item.content}}</view>
    </view>
  </view>
</view>

<!-- <view class="click" :data-id="item.id" @click="click"></view> -->
<navigator url="../chat/chat" open-type="switchTab" class="click" :data-id="item.id" @click="click"></navigator>
</view>
```

```
methods: {
  click(e) {
    console.log(this.chatarr[e.target.dataset.id].isnew);
    this.chatarr[e.target.dataset.id].isnew=0
  },
  click2(e) {
    console.log(this.navarr[e.target.dataset.id].isnew);
    this.navarr[e.target.dataset.id].isnew=0
  }
}
```

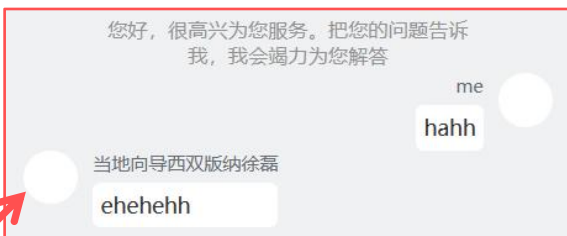

设计过程

消息发送

v-model=""

在表单输入元素或组件上创建双向绑定

消息



```
<script>
export default {
  data() {
    return {
      message: ''
    }
  }
}
</script>

<template>
  <p>Message is: {{ message }}</p>
  <input v-model="message" placeholder="edit m">
</template>
```

```
<view class="container3">
  <view class="iptbox">
    <input type="text" class="ipt" v-model="myipt">
    <image src="/static/message/emoji.png" mode="" class="emoji"></image>
  </view>
  <view class="sendbox" @click="send">
    <image src="/static/message/send.png" mode="" class="send"></image>
  </view>
</view>
</view>
</template>

export default {
  data() {
    return {
      myipt: "",

```

```
methods: {
  send() {
    let getipt = this.myipt;
    this.myipt = "";
    let myinput = {
      ismine: 1,
      msg: getipt
    }
    this.chatarr.push(myinput);
    this.position = (this.chatarr.length+1-9)*50.2
    // this.position = this.chatarr.length+10000
  }
}
```

获取输入数据

输入置空

2 加入数组，发送我的输入

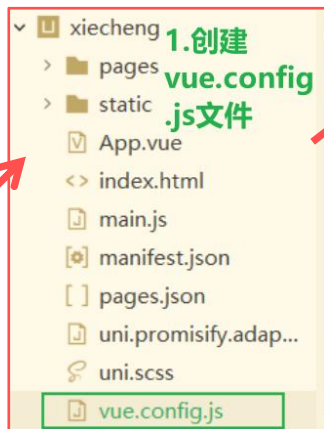
3 定位到底部

消息+Chat

设计过程

消息回复

消息



uni.request(OBJECT)

发起网络请求。

url 开发者服务器接口地址

data 请求的参数

success 收到开发者服务器

成功返回的回调函数

成员分工

消息+Chat

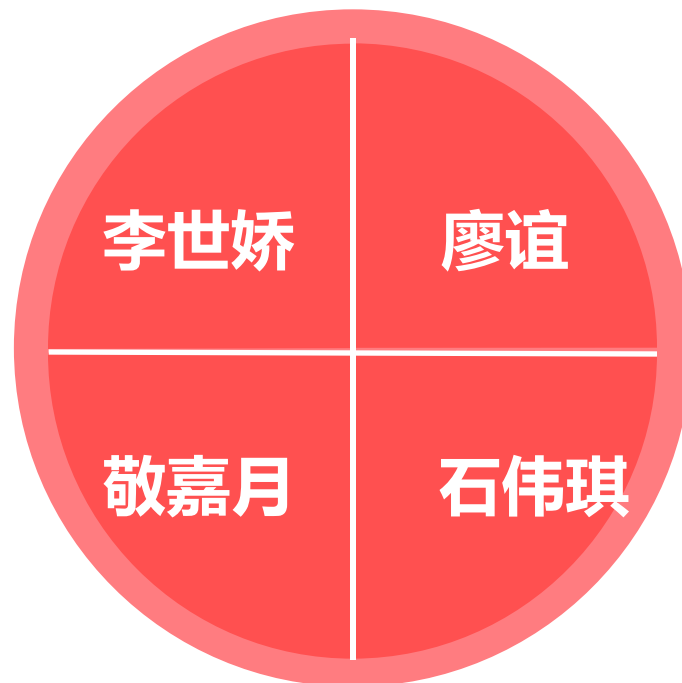
界面设计：消息页面及其子页面

交互设计：对话功能、跳转功能、可滚动视图

社区

界面设计：社区页面及其子页面

交互设计：跳转功能、可滚动视图、定位菜单栏



首页

界面设计：首页及其子页面

交互设计：跳转功能、轮播图、弹窗提示、显示消息提示框

我的

界面设计：个人页面及其子页面

交互设计：跳转功能、登录功能、弹窗提示、显示消息提示框

设计过程

界面设计： 首页及其子页面

交互设计： 跳转功能、轮播图、弹窗提示、显示消息提示框



首页



社区



消息



我的

设计过程



参考页面



效果页面



设计过程



搜索区域



input组件 placeholder占位符

悬浮区域



position定位 fixed固定定位

原来位置不存在，始终相对于模拟器进行偏移

image图像组件 src引入图片地址

相对地址：./表示当前文件所在的文件夹

../表示当前文件所在文件夹的上一级文件夹

/若放在首位，表示根目录

设计过程

分类区域



display:flex 弹性盒子 让子元素在主轴方向上一行显示
flex-direction:column 主轴方向为垂直方向
align-items:center 交叉轴的对齐方式（居中）
justify-content:center 主轴上的对齐方式（居中）

首页



overflow:hidden 隐藏角
position:absolute 绝对定位
top 设置偏移量
原来位置不存在，若父级或祖先元素存在定位，则相对于其最近的进行偏移。
flex-direction:row 主轴方向为水平方向

设计过程



swiper滑块视图容器
子元素只能是 swiper-item
indicator-dots 是否显示面板指示点
indicator-active-color 当前选中的指示点颜色
autoplay interval 自动切换及时间间隔

@click="getFood" 事件绑定
navigator url="" 链接组件
url链接地址后面不能有.vue后缀

设计过程

跳转区域



首页

```
methods: {  
  getFood(){  
    //显示正在加载提示框  
    uni.showLoading({  
      title:"即将呈现"  
    })  
    //定时器  
    setTimeout(function(){  
      //关闭正在加载提示框  
      uni.hideLoading()  
      //显示消息提示框  
      uni.showToast({  
        title:"加载成功",  
        icon:"success",  
      })  
    },1000)  
  }  
}
```

uni.showLoading需要调用uni.hideLoading才会消失
定时器setTimeout(function() {},10);
第一个参数代表执行函数
第二个参数代表多少毫秒后执行前面函数

设计过程



scroll-view 可滚动区域
scroll-x 允许横向滚动
控制滚动条消失在App.vue文件中添加属性.
@click="getS" 事件绑定



```
data() {  
  return {  
    v: "fa",  
    arr: [  
      {  
        name: "上海王宝和大酒店上海餐厅",  
        why: "主打蟹宴的百年老店，中式装修华丽气派",  
        price: "¥ 394"  
      },  
      {  
        name: "大壶春(四川中路店)",  
        why: "坚持做全发面生煎，彰显传统老店品质",  
        price: "¥ 28"  
      },  
      {  
        name: "杏花楼(福州路总店)",  
        why: "创于咸丰元年的老店，各式糕点水准上乘",  
        price: "¥ 260"  
      }  
    ]  
  }  
},  
methods: {  
  getS(){  
    this.v="fa2"  
  }  
}
```

v-bind数据绑定 v-bind:class="v"
v-for列表渲染 v-for="(item,index) in 数组"
item循环数组里的每一项，index数组下标
代码：

<view v-bind:class="v" v-for="(item,index) in arr">店名{{item.name}}.....>

设计过程



navigator url="" 链接组件



@click="getC" 绑定事件
uni.showToast消息提示框

设计过程

广告区域



space-between 两端对齐

首页

```
<!-- 广告区域 -->
<view class="ad-banner-wrap">
  <view class="img_wrap">
    <image src="../../../static/demo1/提示.jpg" class="img1"></image>
  </view>
</view>
<!-- 选项区域 -->
<view class="toolbox">
  <view class="tool-gruop">
    <view class="tool-link">
      <image src="../../../static/demo1/3.1电话.png" class="img2"></image>
      <view class="t">电话预订</view>
    </view>
    <view class="tool-link">
      <image src="../../../static/demo1/下载.png" class="img2"></image>
      <view class="t">下载携程</view>
    </view>
  </view>
  <image src="../../../static/demo1/signin-d.png" class="sign"></image>
</view>
```

设计过程

消息+Chat

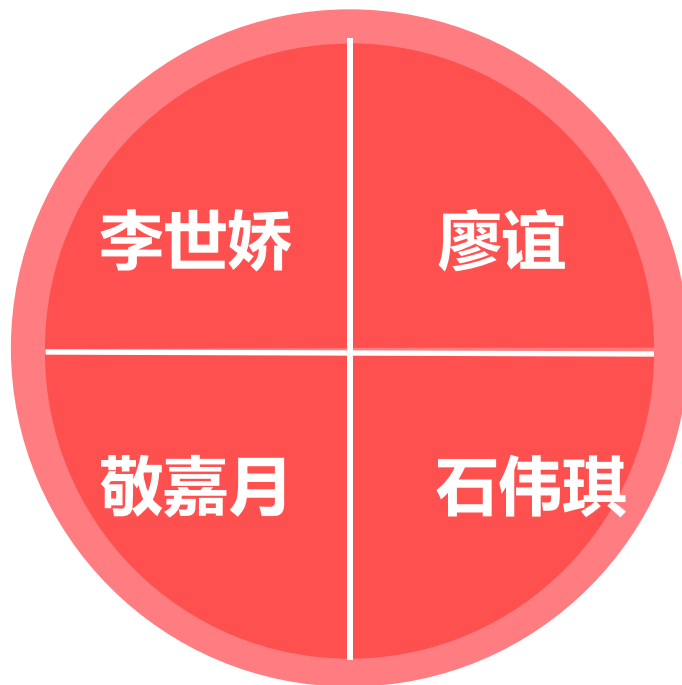
界面设计：消息页面及其子页面

交互设计：对话功能、跳转功能、可滚动视图

社区

界面设计：社区页面及其子页面

交互设计：跳转功能、可滚动视图、定位菜单栏



首页

界面设计：首页及其子页面

交互设计：跳转功能、轮播图、弹窗提示、显示消息提示框

我的

界面设计：个人页面及其子页面

交互设计：跳转功能、登录功能、弹窗提示、显示消息提示框

设计过程

界面设计：社区页面及其子页面
交互设计：跳转功能、可滚动视图、定位菜单栏



首页



社区



消息



我的

设计过程

社区

社区

社区页面能提供给用户提供娱乐选择，能够根据定位推荐附近的地方，其页面设计首先分上下各一个盒子，给他们使用`display:flex`属性使他们的子元素水平分布。

上方盒子点击全球可以跳转到定位页面，使用了`@click`点击事件和在`methods`里使用`uni.navigateTo`方法，右侧使用了`input`输入框，输入框显示的文字使用`placeholder`属性。

下方大盒子左右侧都使用了`scroll-view`组件，设置属性`scroll-y`，使其都能沿`y`轴方向滑动，然后设置`scroll-into-view="id"`，使其滚动到`id`所在元素（给左侧每一个盒子添加一个`@click="getid"`事件并且给其各自的`data-id`，右侧每一个盒子都给他们设置`id`，使其与左侧盒子一一对应，最后在`methods`里写明`getid(e)`方法，使`this.id=e.target.dataset.id`最终实现点击左侧某一盒子，右侧自动滑动到相应部分。设置`scroll-with-animation`让元素在滚动时使用动画过渡。

全球^

目的地/景点/话题...

发现

徒步

避暑

看海

玩乐

酒店

美食



<发现>森林卧室做秋千，是要飞出天际的感觉没错啦

探店达人



查看更多-->



<徒步>徒步探洞，温泉溯溪，正是户外好季节！



首页



社区



chat



消息



我的

设计过程

社区



右侧每一个大盒子都使用了相同的布局，以“看海”为例，从上往下分了4个部分，顶部的图片设置了border-radius属性10%,使图片有一定的圆角。

第三个盒子使用display:flex属性，子元素水平排布，头像使用border-radius50%使其为圆形。

最后一个盒子为其设置了点击事件，使用API：uni.navigateTo跳转到子页面。

设计过程



跳转的两个子页面：



左上角的<都可以点击使页面返回到社区主页面，都使用的是uni.switchTab.

子页面主要是页面布局
定位页面的热门城市里使用了属性justify-content: space-between，使其在一行内均匀分散排布。
详情页面可以向下滑动看更多。

成员分工

消息+Chat

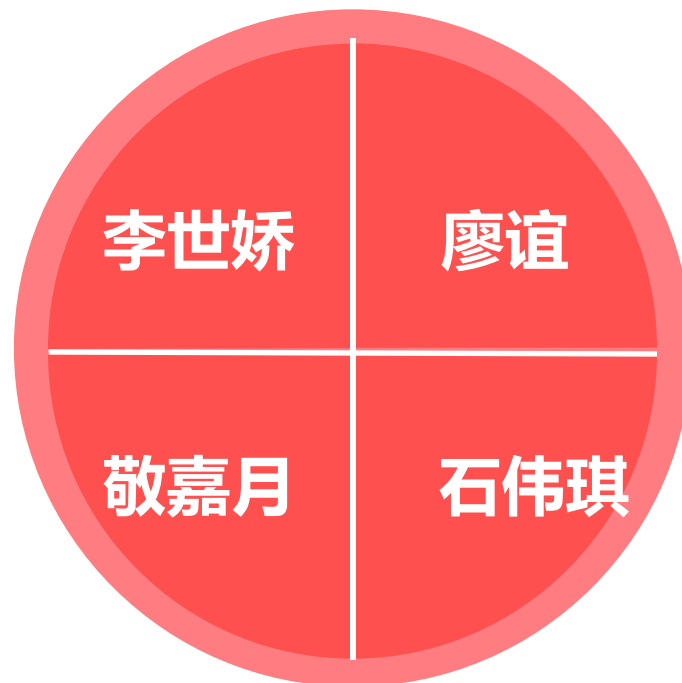
界面设计：消息页面及其子页面

交互设计：对话功能、跳转功能、可滚动视图

社区

界面设计：社区页面及其子页面

交互设计：跳转功能、可滚动视图、定位菜单栏



首页

界面设计：首页及其子页面

交互设计：跳转功能、轮播图、弹窗提示、显示消息提示框

我的

界面设计：个人页面及其子页面

交互设计：跳转功能、登录功能、弹窗提示、显示消息提示框

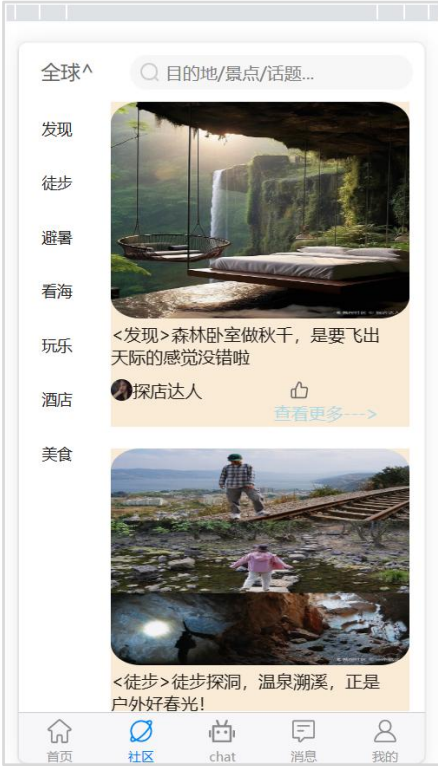
设计过程

界面设计：个人页面及其子页面

交互设计：跳转功能、登录功能、弹窗提示、显示消息提示框



首页



社区



消息



我的

设计过程



布局:

视图容器nav:

设置display属性, 主轴对齐方式为justify-content: space-between;

使两个图标沿主轴均匀排列, 空白间距均分在弹性盒子两侧。

视图容器nav2:

分为视图容器textview和btg两个子级;

btg布局模式为弹性盒子, 主轴对齐方式为justify-content: center; 侧轴对齐方式为align-items: center;

使两个button组件沿主轴和侧轴居中排列。并设置text-align: center属性, 使行内文字居中。

设置边界属性border: 2px solid white;使btn1产生一个实线的边框, background-color设置为 transparent透明

视图容器nav3:

分为三个部分

每个部分布局模式都为弹性盒子

细分弹性盒子, 每个父盒子设为flex布局, 子元素为一个图标和一段文字, 根据页面需求设置属性

如设置主轴方向为: flex-direction: column;从上到下

在大视图容器中设置背景属性background-image:

url(/static/login/back.png);





API的使用（界面交互反馈与数据缓存）：

- 1.显示提示框：使用uni.showModal，显示模态弹窗，如果用户点击确定，confirm字段将会返回true，提示框显示登录成功，如果点击取消将会显示登录失败。
- 2.显示正在加载的提示框：使用uni.showLoading
- 3.关闭提示框：使用定时器setTimeout(function() {}, 10)，并调用uni.hideLoading（隐藏loading提示框。）
- 4.关闭正在加载提示框后设置头像和昵称
// 生命周期
- 5.在页面加载的时候获取登录缓存：使用uni.getStorage从本地缓存中,通过一个key值获取指定 key 对应的内容。，通过success参数判断是否存在缓存，如果存在将会设置图片或者文字。

这样就达到了点击button实现登录的

设计过程



<

手机快捷登录

请输入手机号

请输入密码

登 录

短信验证码登录 找回密码

☒ 阅读并同意携程的<服务协议>和<个人信息保护政策>

布局:

input输入标签的type属性

文本框type=text

svg画线

密码框的定义type=password

svg画线

定义placeholder属性提供提示信息hint

navigator页面跳转

radio定义单选按钮

四

总结收获

总结收获

通过本学期uniapp的学习巩固了前端Vue.js框架的知识，在学习过程中通过查找官方文档解决学习中遇到的一些问题，有效地提高了自学能力，学会了使用不同组件设计页面，实现界面跳转、消息弹窗、提示信息等交互功能，在独立编写一套代码的过程中训练了逻辑思维能力。

谢谢老师指导和小伙伴们 们的负责认真





海量视频 贴身学习



超多干货 实时更新

THANKS

— 谢谢 —