

实验主要目的：

用于手机的个人日常管理系统的设计与开发，实现个人收支管理、日程管理、日程数据库的添删改查等功能，本次使用微信小程序开发，使得多平台手机能够共同使用此软件，并且小程序不用额外下载 APP，占用手机的空间。本次服务端部署在本地，要进入其局域网才能实现功能，以后可以考虑部署到云端。

日程管理是将每天计划、工作和事务安排在日期中，并做一个有效的记录，方便管理日程的工作和事务，达到工作备忘的目的。

让用户在移动终端上体验方便、快捷的个人助理管理。满足用户将学习工作计划、会议安排等事务随时记录到此系统中，并且可以方便得实现收支、日程记录的查找、添加、修改，删除。显示当前天的日程信息。

有日记功能，可以让用户添加位置信息，图片信息和文字信息，使得这三者数据共存。日记可以实现查询添加删除功能。

有投资功能，此功能可以帮助用户更好地管理自己的资产，记录时间，自动计算每天的收益等。

首页有实时天气更新，方便用户查看实时天气。

二、实验主要内容：

1) 需求分析：

用微信小程序实现个人助理，实现日程管理，投资管理，收支管理，日记管理，实现和服务端的交互功能，服务端完成对数据库的操作。

2) 详细设计

前端：微信小程序

服务端：springboot+mybatis 架构

数据库:mysql

数据库结构：

```
/*
Navicat MySQL Data Transfer

Source Server        : cjw
Source Server Version : 50615
Source Host          : localhost:3306
Source Database       : personal_assistant

Target Server Type    : MYSQL
Target Server Version : 50615
File Encoding         : 65001
```

Date: 2019-07-15 03:06:18

*/

SET FOREIGN_KEY_CHECKS=0;

-- -----
-- Table structure for tbl_expenses
-- -----

```
DROP TABLE IF EXISTS `tbl_expenses`;  
CREATE TABLE `tbl_expenses` (  
  `exp_id` int(20) NOT NULL AUTO_INCREMENT,  
  `exp_money` int(200) NOT NULL,  
  `exp_type` varchar(50) NOT NULL,  
  `exp_name` varchar(50) DEFAULT NULL,  
  `exp_note` varchar(500) DEFAULT NULL,  
  `exp_date` date NOT NULL,  
  `user_id` int(200) NOT NULL,  
  PRIMARY KEY (`exp_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;
```

-- -----
-- Records of tbl_expenses
-- -----

```
INSERT INTO `tbl_expenses` VALUES ('2', '100', 'test', null, null,  
'2019-07-13', '1');  
INSERT INTO `tbl_expenses` VALUES ('4', '0', '饮食', '', null, '2019-07-  
14', '11');
```

-- -----
-- Table structure for tbl_income
-- -----

```
DROP TABLE IF EXISTS `tbl_income`;  
CREATE TABLE `tbl_income` (  
  `inc_id` int(20) NOT NULL AUTO_INCREMENT,  
  `inc_money` int(200) NOT NULL,  
  `inc_type` varchar(50) NOT NULL,  
  `inc_name` varchar(50) DEFAULT NULL,  
  `inc_note` varchar(500) DEFAULT NULL,  
  `inc_date` date NOT NULL,  
  `user_id` int(20) NOT NULL,  
  PRIMARY KEY (`inc_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;
```

-- -----

```

-- Records of tbl_income
-- -----
INSERT INTO `tbl_income` VALUES ('2', '200', 'test', null, null, '2019-07-13', '1');
INSERT INTO `tbl_income` VALUES ('3', '300', 't2', null, null, '2019-07-12', '1');
INSERT INTO `tbl_income` VALUES ('5', '0', '工资', '', null, '2019-07-15', '11');

-- -----
-- Table structure for tbl_investment
-- -----
DROP TABLE IF EXISTS `tbl_investment`;
CREATE TABLE `tbl_investment` (
  `inv_id` int(20) NOT NULL AUTO_INCREMENT,
  `inv_name` varchar(50) NOT NULL,
  `inv_money` double(50,0) NOT NULL,
  `inv_profit_money` double(50,2) DEFAULT NULL,
  `inv_rate` double(10,2) NOT NULL,
  `inv_start_date` date NOT NULL,
  `inv_end_date` date DEFAULT NULL,
  `user_id` int(20) NOT NULL,
  PRIMARY KEY (`inv_id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

-- -----
-- Records of tbl_investment
-- -----
INSERT INTO `tbl_investment` VALUES ('2', '123', '5000', '250.00', '0.05', '2019-07-14', null, '11');
INSERT INTO `tbl_investment` VALUES ('3', '123', '5000', '1077.53', '0.05', '2019-07-10', null, '1');
INSERT INTO `tbl_investment` VALUES ('5', '', '666', '0.00', '0.50', '2019-07-15', null, '11');

-- -----
-- Table structure for tbl_invtype
-- -----
DROP TABLE IF EXISTS `tbl_invtype`;
CREATE TABLE `tbl_invtype` (
  `inv_name_id` int(20) NOT NULL AUTO_INCREMENT,
  `inv_name` varchar(20) NOT NULL,
  `inv_money_count` double(20,2) NOT NULL,
  `inv_pfmoney_count` double(20,2) NOT NULL,

```

```

`user_id` int(20) NOT NULL,
PRIMARY KEY (`inv_name_id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

-- -----
-- Records of tbl_invtype
-- -----
INSERT INTO `tbl_invtype` VALUES ('1', '余额宝', '100.00', '10.00',
'11');

-- -----
-- Table structure for tbl_photodiary
-- -----
DROP TABLE IF EXISTS `tbl_photodiary`;
CREATE TABLE `tbl_photodiary` (
  `phd_id` int(20) NOT NULL AUTO_INCREMENT,
  `phd_diary` varchar(100) NOT NULL,
  `phd_photo` mediumtext NOT NULL,
  `phd_position` varchar(100) NOT NULL,
  `phd_date` date NOT NULL,
  `user_id` int(20) NOT NULL,
  PRIMARY KEY (`phd_id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

-- -----
-- Records of tbl_photodiary
-- -----
INSERT INTO `tbl_photodiary` VALUES ('2', '天气真好', 'img/google-earth-
view-1140.jpg', '789', '2019-07-13', '1');
INSERT INTO `tbl_photodiary` VALUES ('3', '天气真好', 'img/google-earth-
view-1140.jpg', '7', '2019-07-13', '11');
INSERT INTO `tbl_photodiary` VALUES ('4', '天气真好', 'img/google-earth-
view-1447.jpg', '123', '2019-01-01', '11');

-- -----
-- Table structure for tbl_schedule
-- -----
DROP TABLE IF EXISTS `tbl_schedule`;
CREATE TABLE `tbl_schedule` (
  `sc_id` int(20) NOT NULL AUTO_INCREMENT,
  `sc_name` varchar(50) NOT NULL,
  `sc_start_time` time(5) NOT NULL,
  `sc_end_time` time(5) DEFAULT NULL,
  `sc_start_date` date NOT NULL,

```

```

`sc_end_date` date DEFAULT NULL,
`sc_repeat_type` varchar(20) DEFAULT NULL,
`user_id` int(20) NOT NULL,
`sc_repeat_times` int(20) DEFAULT NULL,
`sc_remind_days` varchar(255) DEFAULT NULL,
PRIMARY KEY (`sc_id`)
) ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=utf8;

-- -----
-- Records of tbl_schedule
-- -----
INSERT INTO `tbl_schedule` VALUES ('12', '11', '10:00:00.00000', null,
'2019-07-15', '2019-07-28', '每天重复', '11', '1', null);
INSERT INTO `tbl_schedule` VALUES ('13', '11', '10:00:00.00000', null,
'2019-07-15', '2019-07-26', '每天重复', '11', '1', null);
INSERT INTO `tbl_schedule` VALUES ('14', '1', '10:00:00.00000', null,
'2019-07-15', '2019-07-26', '每天重复', '11', '1', '2019-07-151,');
INSERT INTO `tbl_schedule` VALUES ('16', '取快递', '12:00:00.00000',
null, '2019-07-16', '2019-07-18', '每天重复', '11', '2', '2019-07-
function toString() { [native code] },2019-07-function toString()
{ [native code] },');
INSERT INTO `tbl_schedule` VALUES ('17', '快递', '10:00:00.00000', null,
'2019-07-16', '2019-07-17', '每天重复', '11', '2', '2019-07-17,2019-07-
18,');
INSERT INTO `tbl_schedule` VALUES ('18', '打印', '10:00:00.00000', null,
'2019-07-16', '2019-07-16', '每天重复', '11', '1', '2019-07-17,');
INSERT INTO `tbl_schedule` VALUES ('19', '1', '10:00:00.00000', null,
'2019-07-13', '2019-07-17', '每天重复', '11', '5', '2019-07-19,2019-07-
20,2019-07-21,2019-07-22,2019-07-23,');

-- -----
-- Table structure for tbl_user
-- -----
DROP TABLE IF EXISTS `tbl_user`;
CREATE TABLE `tbl_user` (
  `user_id` int(20) NOT NULL AUTO_INCREMENT,
  `user_name` varchar(20) NOT NULL,
  `exp_count` int(100) DEFAULT NULL,
  `inc_count` int(100) DEFAULT NULL,
  `inv_count` double(100,2) DEFAULT NULL,
  `inv_pfcoun` double(100,2) DEFAULT NULL,
  PRIMARY KEY (`user_id`)
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=utf8;

```

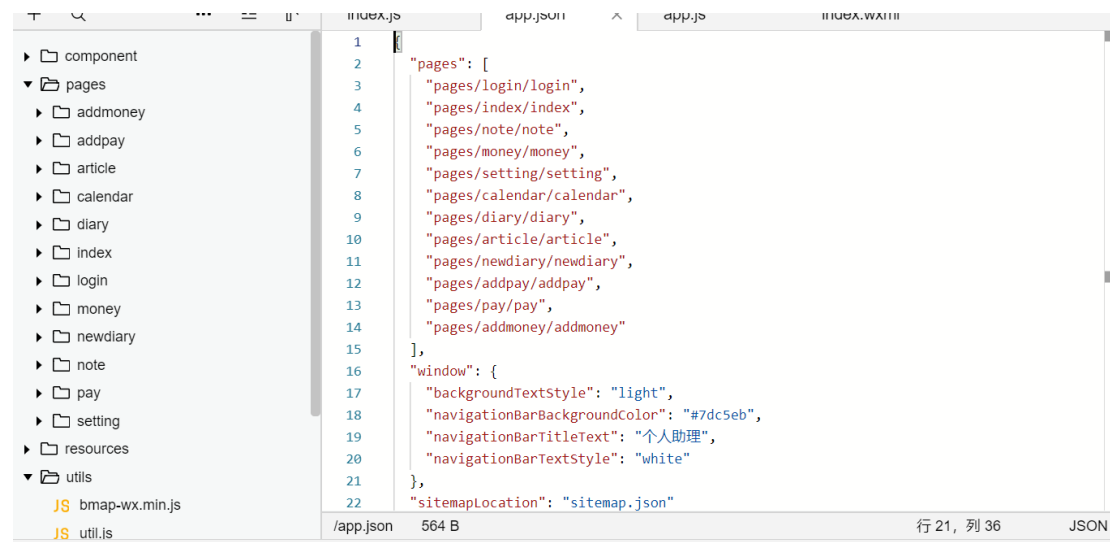
```

-- -----
-- Records of tbl_user
-- -----

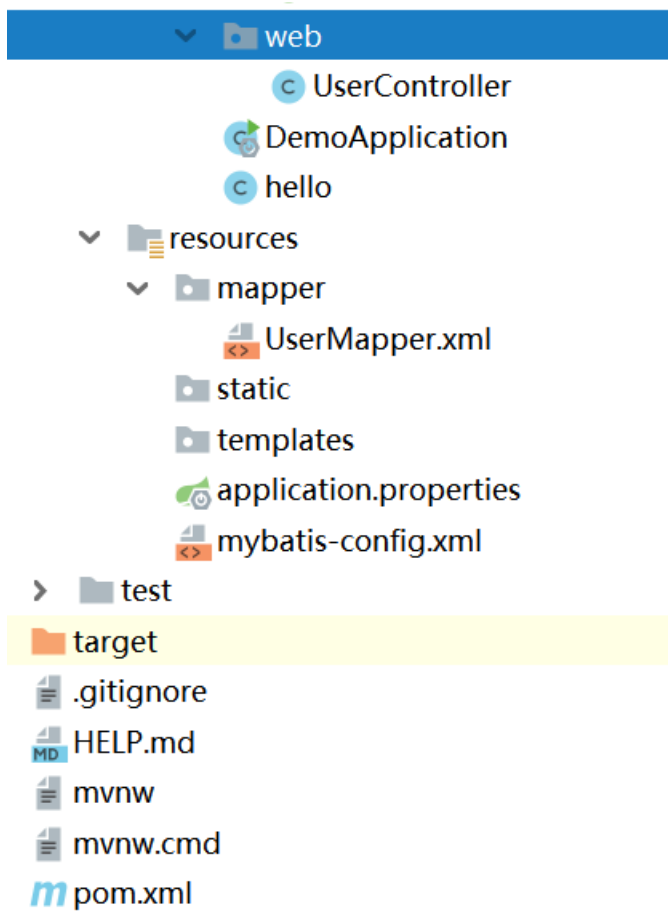
INSERT INTO `tbl_user` VALUES ('4', '123', null, null, null, null);
INSERT INTO `tbl_user` VALUES ('5', '456', null, null, null, null);
INSERT INTO `tbl_user` VALUES ('10', '467', null, null, null, null);
INSERT INTO `tbl_user` VALUES ('11', '湛泸', null, null, '100.00',
'100.00');

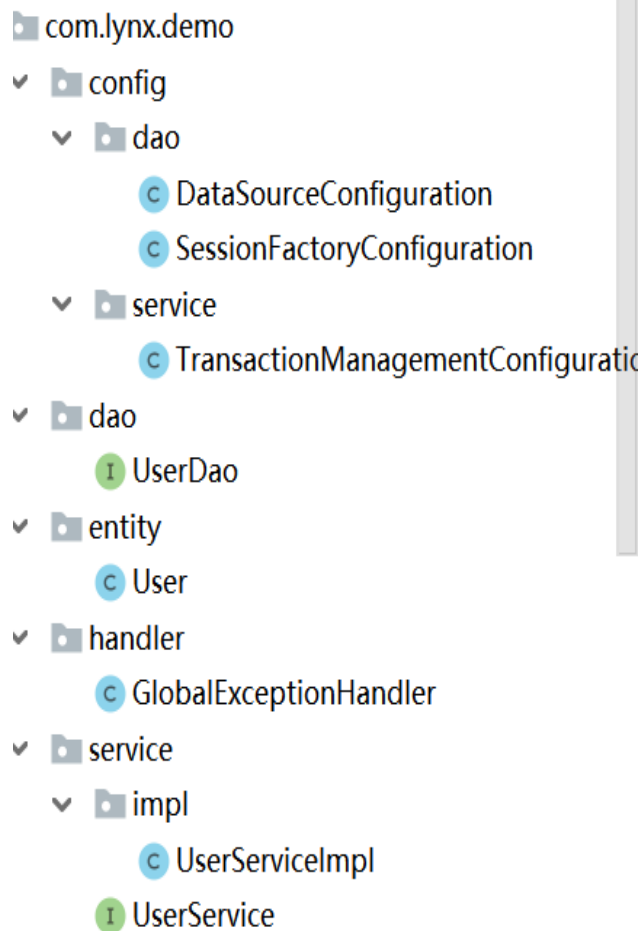
```

小程序结构



服务端结构:





3) 小程序编码（编码较多，就不一一贴上了，详情见源码）

index.wxml

```
<view>
  <view class="page-bottom">
    <view class="page-content">
      <view class="wc">
        <navigator url="../index/index" hover-class="navigator-hover">
          <image src="img/home.png" style="width:20px ; height:20px"></image>
          主页</navigator>
        </view>
        <view class="wc">
          <navigator url="../calendar/calendar" hover-class="navigator-hover">
            <image src="img/calendar-alt.png" style="width:20px ;
height:20px"></image>
            日程安排</navigator>
          </view>
          <view class="wc">
            <navigator url="../diary/diary" hover-class="navigator-hover">
              <image src="img/diary.png" style="width:20px ; height:20px"></image>
              日记</navigator>
            </view>
          </view>
        </view>
      </view>
    </view>
  </view>
```



```

        <view class="wc">
            <navigator url="../pay/pay" hover-class="navigator-hover">
                <image src="img/feather.png" style="width:20px ;
height:20px"></image>
                收支记账</navigator>
            </view>
            <view class="wc">
                <navigator url="../money/money" hover-class="navigator-hover">
                    <image src="img/chart-areaspline.png" style="width:20px ;
height:20px"></image>
                    投资分析</navigator>
                </view>
                <view class="wc">
                    <navigator url="../setting/setting" hover-class="navigator-hover">
                        <image src="img/settings.png" style="width:30px ;
height:30px"></image>
                        设置</navigator>
                    </view>
                </view>
            </view>
        </view>

```

```

        <scroll-view scroll-y="true" style="height:700px" bindtouchmove="tap_drag"
bindtouchend="tap_end" bindtouchstart="tap_start" class="page-top {{open ? 'c-
state1' : ''}}">

```

```

        <loading hidden="{{loadingStatus}}">
            加载中...
        </loading>
        <view wx:if="{{userLocation == 3}}" class="box weather_bg list_left">
            <text class='tips' bindtap='refreshCurrentPage'>
                定位异常，无法正确获取您的定位信息\n
                1.请重新刷新页面\n
                2.尝试换个地方重新刷新页面\n
                3.晚些时候再刷新页面\n
                4.非中国境内，无法正确获取\n
            </text>
        </view>
        <view wx:if="{{userLocation == 2}}" class="box city_title list_center">
            <text class='tips' bindtap='refreshCurrentPage'>
                没有授权获取定位信息，功能无法使用\n
                确认授权后，请下拉或者点击文字刷新获取数据
            </text>
            <button class='btn' bindtap='gotoAuthPage'>点击开启授权</button>
        </view>

```

```

<view wx:elif="{{userLocation == 1}}">

    <view style="margin-left:10px;margin-top:40px;display: flex;flex-direction: row;">
天气情况
    <view style="margin-left:50px">{{cityInfo.addressComponent.city}}</view>
    </view>

    <view class="box weather_info">
        <view class='f_red'>{{weatherInfo.date}}</view>
        <view>
            <text class='f_black'>{{weatherInfo.weatherDesc}}</text>
            <text class='ml20'>{{weatherInfo.temperature}}</text>
            <text class='ml20'>{{weatherInfo.wind}}</text>
        </view>
        <view>PM2.5: {{weatherInfo.pm25}}</view>
    </view>

</view>

```

```

<image bindtap="tap_ch" src="img/menu.png"></image>

<view style="margin-top:10px;margin-left:10px;margin-bottom:50px;">

    <view class="activity">
    <text>投资总收益</text>
    <text style="margin-left:200px">{{sum}}</text>
    </view>
    <view class="activity">
    <text>投资总金额</text>
    <text style="margin-left:200px">{{sumcount}}</text>
    </view>
    <view class="activity">
    <text>总收入</text>
    <text style="margin-left:250px">{{sumincome}}</text>
    </view>
    <view class="activity">
    <text>总支出</text>

```

```

        <text style="margin-left:250px">{{sumoutcome}}</text>
      </view>

    </view>
  </scroll-view>
</view>

```

index.js

```

const app = getApp();
var util = require('../../utils/util.js');
var bmap = require('../../utils/bmap-wx.min.js');
Page({
  data: {
    loadingStatus: false,
    userLocation: 0, // 0 默认没有状态, 1 获取到用户定位, 2 用户拒绝提供定位信息, 3 无法获取正确定位
    cityInfo: {}, //城市信息
    weatherInfo: {}, //天气信息
    currentTime: '', // 当前时间
    open: false,
    mark: 0,
    newmark: 0,
    istoright: true,
    userInfo: {},
    sum: '',
    sumcount: '',
    sumincome: '',
    sumoutcome: '',
    hasUserInfo: false,
    canIUse: wx.canIUse('button.open-type.getUserInfo'),
    id: '',
    notify: [
      // {
      //   title: '取快递'
      // }
    ],
    income: [
      // {
      //   money: 100,
      //   info: 111,
      //   type: '其他'
      // }
    ],
  },

```

```

invest: [
  // {
  //   startdate: '2019-07-01',
  //   enddate: '2019-08-17',
  //   money: 10000,
  //   rate: 0.02,
  //   type: '银行存款',
  //   profit: '2'
  // }
]

},

//获取提醒
onShow: function () {
  var that = this;
  console.log(getApp().globalData.userInfo.nickName);
  wx.request({
    url: "http://192.168.43.58:8080/personalAssistant/user/addUser",
    data: { "userName": getApp().globalData.userInfo.nickName },
    method: 'POST',
    success: function (res) {
      var id = res.data.userId;
      if (id == '') {
        var toastText = '获取数据失败' + res.data.errMsg;
        wx.showToast({
          title: toastText,
          icon: '',
          duration: 2000
        });
      } else {
        console.log("2", getApp().globalData);
        getApp().globalData.userId = id;
        that.data.id = id;
        console.log("3", getApp().globalData);

        wx.request({
          url:
"http://192.168.43.58:8080/personalAssistant/investment/updateInvestment",
          data: { "userId": getApp().globalData.userId },
          method: 'GET',
          success: function (res) {
            console.log(getApp().globalData.userId);
          }
        })
      }
    }
  });
}

```

```

});

wx.request({
  url: "http://192.168.43.58:8080/personalAssistant/user/sumInvPf",
  data: { "userId": getApp().globalData.userId },
  method: 'GET',
  success: function (res) {
    var sum = res.data.sumInvPf;
    if (sum == null) {
      var toastText = '获取数据失败' + res.data.errMsg;
      wx.showToast({
        title: toastText,
        icon: '',
        duration: 2000
      });
    } else {
      that.setData({
        sum: sum
      });
    }
  }
});

wx.request({
  url: "http://192.168.43.58:8080/personalAssistant/user/sumInv",
  data: { "userId": getApp().globalData.userId },
  method: 'GET',
  success: function (res) {
    var sum = res.data.sumInv;
    if (sum == null) {
      var toastText = '获取数据失败' + res.data.errMsg;
      wx.showToast({
        title: toastText,
        icon: '',
        duration: 2000
      });
    } else {
      that.setData({
        sumcount: sum
      });
    }
  }
});

```

```

wx.request({
  url: "http://192.168.43.58:8080/personalAssistant/user/sumIncome",
  data: { "userId": getApp().globalData.userId },
  method: 'GET',
  success: function (res) {
    var sum = res.data.sumIncome;
    if (sum == null) {
      var toastText = '获取数据失败' + res.data.errMsg;
      wx.showToast({
        title: toastText,
        icon: '',
        duration: 2000
      });
    } else {
      that.setData({
        sumincome: sum
      });
    }
  }
});

wx.request({
  url: "http://192.168.43.58:8080/personalAssistant/user/sumExcepses",
  data: { "userId": getApp().globalData.userId },
  method: 'GET',
  success: function (res) {
    var sum = res.data.sumExcepses;
    if (sum == null) {
      var toastText = '获取数据失败' + res.data.errMsg;
      wx.showToast({
        title: toastText,
        icon: '',
        duration: 2000
      });
    } else {
      that.setData({
        sumoutcome: sum
      });
    }
  }
});
}
});

```

```

},

updateUserLocation: function (v) {
  this.setData({
    loadingStatus: true, // 加载完成, 隐藏加载动画
    userLocation: v // 更新当前页面的变量值
  });
},

// 跳转到授权页面
gotoAuthPage: function () {
  wx.openSetting({
    fail: function () {
      wx.showModal({
        title: '操作失败',
        content: '请到微信中自行设置授权操作',
        confirmText: '知道了',
        confirmColor: '#345391',
        showCancel: false
      })
    }
  });
},

// 刷新当前页面 (相当于下拉刷新页面)
refreshCurrentPage: function () {
  wx.startPullDownRefresh({});
},

// 获取地理位置和天气的方法
getLocationAndWeather: function () {
  var that = this;
  // 使用百度地图提供的 ak 标记, 初始化地图数据
  var BMap = new bmap.BMapWX({
    ak: app.globalData.bmap_ak
  });
  // 失败执行的方法
  var fail = function (data) {
    if (data.errMsg == 'getLocation:fail auth deny') {
      // 更新页面状态为 2, 用户拒绝提供位置信息
      that.updateUserLocation(2);
    } else {

```

```

        // 更新页面状态为 3，获取定位信息错误，或者定位不在大陆境内
        that.updateUserLocation(3);
    }
};
// 获取城市信息成功执行的方法
var regeocoding_success = function (data) {
    // 获取地址数据
    var city_info = data.originalData.result;
    // 二次处理数据，将经纬度的数据，保留四舍五入保留五位小数
    city_info.location.lat = city_info.location.lat.toFixed(5);
    city_info.location.lng = city_info.location.lng.toFixed(5);
    // 更新页面状态为 1，用户允许授权定位信息
    that.updateUserLocation(1);
    // 将相关数据，写到初始化的 data 数据中
    that.setData({
        cityInfo: city_info
    });
};
// 获取天气成功执行的方法
var weather_success = function (data) {
    // 获取天气基本数据
    var weatherInfo = data.currentWeather[0];
    // 获取天气扩展信息
    var weatherInfoExt = data.originalData.results[0].weather_data;
    // 将天气预报数据写入定义（初始化）的变量中
    that.setData({
        weatherInfo: weatherInfo,
        weatherInfoExt: weatherInfoExt
    });
    // 天气获取成功之后，再调用其他数据（regeocoding 接口有 BUG，没有数据依然返回成功状态）

    // 调用接口获取城市地区信息
    BMap.regeocoding({
        fail: fail,
        success: regeocoding_success
    });
};
// 调用接口获取天气信息
BMap.weather({
    fail: fail,
    success: weather_success
});
},

```



```

onPullDownRefresh: function () {
  wx.showNavigationBarLoading(); // 显示顶部刷新图标
  this.getLocationAndWeather(); // 调用获取天气和地理位置的方法
  wx.hideNavigationBarLoading(); // 隐藏导航栏加载框
  wx.stopPullDownRefresh(); // 停止下拉动作
},

onLoad: function () {

  app.checkUserLocation();
  // 获取当前时间，并拆分成日期和时间两部分
  var current_time = util.formatTime(new Date()).split(' ');
  // 将时间部分记录，用于判断早晚
  this.setData({
    currentTime: current_time[1]
  });
  // 调用获取天气和地理位置的方法
  this.getLocationAndWeather();
  var that = this;
  wx.request({
    url:
"http://192.168.43.58:8080/personalAssistant/investment/updateInvestment",
    data: { "userId": that.data.id },
    method: 'GET',
    success: function (res) {
      console.log(that.data.id);
    }
  });
},

tap_ch: function (e) {
  if (this.data.open) {
    this.setData({
      open: false
    });
  } else {
    this.setData({
      open: true
    });
  }
},

tap_start: function (e) {
  // touchstart 事件
  this.data.mark = this.data.newmark = e.touches[0].pageX;

```

```

    },
    tap_drag: function (e) {
        // touchmove 事件
        /*
         * 手指从左向右移动
         * @newmark 是指移动的最新点的 x 轴坐标 , @mark 是指原点 x 轴坐标
         */
        this.data.newmark = e.touches[0].pageX;
        if (this.data.mark < this.data.newmark) {
            this.istoright = true;
        }
        /*
         * 手指从右向左移动
         * @newmark 是指移动的最新点的 x 轴坐标 , @mark 是指原点 x 轴坐标
         */
        if (this.data.mark > this.data.newmark) {
            this.istoright = false;
        }
        this.data.mark = this.data.newmark;
    },
    tap_end: function (e) {
        // touchend 事件
        this.data.mark = 0;
        this.data.newmark = 0;
        if (this.istoright) {
            this.setData({
                open: true
            });
        } else {
            this.setData({
                open: false
            });
        }
    }
}

})

```

operation.wxml

```

<!--pages/operation.wxml-->
<view class="container">
    <form bindsubmit="formSubmit" bindreset="formReset">

```

```

<view class="row">
  <text>名称: </text>
  <input type="text" name="name" placeholder="请输入设备名称" value="{{name}}" />
</view>
<view class="row">
  <text>类型: </text>
  <input type="text" name="type" placeholder="请输入设备类型" value="{{type}}" />
</view>
<view class="row">
  <text>型号: </text>
  <input type="text" name="model" placeholder="请输入设备型号" value="{{model}}"
/>
</view>
<view class="row">
  <text>信息: </text>
  <input type="text" name="info" placeholder="请输入设备信息" value="{{info}}" />
</view>
<view class="row">
  <button type="default" form-type="submit">提交</button>
  <button type="default" form-type="reset">清空</button>
</view>

</form>
</view>

```

operation.js

//获取应用实例

```
const app = getApp()
```

```

Page({
  data: {
    winWidth: 0,
    winHeight: 0,
    // tab 切换
    currentTab: 0,
    money:0,
    out:0,
    id:app.globalData.userId,
    income:[
      // {
      //   date:'2019-07-01',
      //   money:100,
      //   info:111,
      //   type:'其他'
    ]
  }
})

```

```

        // },
    ],
    outcome: [
        // {
        //     date: '2019-07-01',
        //     money: 100,
        //     info: 111,
        //     type: '饮食'
        // },
    ]
},
onShow:function(){
    var that = this;
    console.log(that.data.id);
    wx.request({
        url:"http://192.168.43.58:8080/personalAssistant/expenses/getExpensesList",
        data: {"userId":that.data.id},
        method: 'GET',
        success: function (res) {
            var list = res.data.list;
            if (list == null) {
                var toastText = '获取数据失败' + res.data.errMsg;
                wx.showToast({
                    title: toastText,
                    icon: '',
                    duration: 2000
                });
            } else {
                for (var index in list) {
                    list[index].expDate=list[index].expDate.substring(0,10)
                }
                that.setData({
                    outcome: list
                });
            }
        }
    });
    wx.request({
        url: "http://192.168.43.58:8080/personalAssistant/income/getIncomeList",
        data: { "userId": that.data.id },
        method: 'GET',
        success: function (res) {
            var list = res.data.list;
            if (list == null) {

```

```

        var toastText = '获取数据失败' + res.data.errMsg;
        wx.showToast({
            title: toastText,
            icon: '',
            duration: 2000
        });
    } else {
        for (var index in list) {
            list[index].incDate = list[index].incDate.substring(0, 10)
        }
        that.setData({
            income: list
        });
    }
}

});

wx.request({
    url: "http://192.168.43.58:8080/personalAssistant/user/sumExcepsses",
    data: { "userId": that.data.id },
    method: 'GET',
    success: function (res) {
        var out = res.data.sumExcepsses;
        if (out == null) {
            var toastText = '获取数据失败' + res.data.errMsg;
            wx.showToast({
                title: toastText,
                icon: '',
                duration: 2000
            });
        } else {
            that.setData({
                out: out
            });
        }
    }
});

wx.request({
    url: "http://192.168.43.58:8080/personalAssistant/user/sumIncome",
    data: { "userId": that.data.id },
    method: 'GET',
    success: function (res) {
        var money = res.data.sumIncome;
        if (money == null) {
            var toastText = '获取数据失败' + res.data.errMsg;

```

```

        wx.showToast({
            title: toastText,
            icon: '',
            duration: 2000
        });
    } else {
        that.setData({
            money: money
        });
    }
}
})
},
onLoad: function () {
    var that = this;
    that.setData({
        id: app.globalData.userId
    });
    /**
     * 获取系统信息
     */
    wx.getSystemInfo({
        success: function (res) {
            that.setData({
                winWidth: res.windowWidth,
                winHeight: res.windowHeight
            });
        }
    });
},
/**
 * 滑动切换 tab
 */
bindChange: function (e) {
    var that = this;
    that.setData({ currentTab: e.detail.current });
},
/**
 * 点击 tab 切换
 */
swichNav: function (e) {
    var that = this;
    if (this.data.currentTab === e.target.dataset.current) {
        return false;
    }
}

```

```

    } else {
      that.setData({
        currentTab: e.target.dataset.current
      })
    }
  },
  clickTab: function (e) {
    var that = this;
    if (this.data.currentTab === e.target.dataset.current) {
      return false;
    } else {
      that.setData({
        currentTab: e.target.dataset.current,
      })
    }
  },
  delete: function (e) {
    var that = this;
    var index = e.currentTarget.dataset.index;
    wx.showModal({
      title: '提示',
      content: '确定要删除此条记录吗? ',
      success: function (res) {
        if (res.confirm) {
          console.log('点击确定了');

          if(that.data.currentTab==0){
            wx.request({
              url: "http://192.168.43.58:8080/personalAssistant/income/deleteIncome",
              data: { incId: that.data.income[e.target.dataset.index].incId },
              method: 'GET',
              success: function (res) {
                var result = res.data.success
                var toastText = "删除成功! ";
                if (result != true) {
                  toastText = "删除失败" + res.data.errMsg;
                } else {
                  that.data.income.splice(e.target.dataset.index, 1)
                  //渲染数据
                  that.setData({
                    income: that.data.income
                  });
                }
              }
            })
            wx.showToast({

```

```

        title: toastText,
        icon: '',
        duration: 2000
    });
    }
    })
    }else{
    wx.request({
        url:
"http://192.168.43.58:8080/personalAssistant/expenses/deleteExpenses",
        data: { expId: that.data.outcome[e.target.dataset.index].expId},
        method: 'GET',
        success: function (res) {
            var result = res.data.success
            var toastText = "删除成功! ";
            if (result != true) {
                toastText = "删除失败" + res.data.errMsg;
            } else {
                that.data.outcome.splice(e.target.dataset.index, 1)
                //渲染数据
                that.setData({
                    outcome: that.data.outcome
                });
            }
            wx.showToast({
                title: toastText,
                icon: '',
                duration: 2000
            });
        }
    })
    }

    } else if (res.cancel) {

    }

    }
    })
    }
    })

```

4) 服务端编码

TblUserMapper.java

```
package com.example.demo.dao;

import com.example.demo.entity.TblSchedule;
import com.example.demo.entity.TblUser;

import java.util.List;

/**
 * <p>
 * Mapper 接口
 * </p>
 *
 * @author c.jw
 * @since 2019-07-11
 */
public interface TblUserMapper {
    TblUser queryUserById(int userId);
    TblUser queryUserByName(String userName);
    int insertUser(TblUser user);
    int updateUser(TblUser user);
    int deleteUser(int userId);
}
```

TblUser.java

```
@Data
public class TblUser {

    /**
     *
     */
    private Integer userId;

    private String userName;

    private Integer expCount;

    private Integer incCount;

    private Double invCount;

    private Double invPfcount;

    public void setUserId(int userId) {
        this.userId = userId;
    }

    public int getUserId() { return this.userId; }
```

UserServiceimpl.java

```

@Service
public class TblUserServiceImpl implements ITblUserService {

    @Autowired
    private TblUserMapper tblUserMapper;
    @Autowired
    private TblIncomeMapper tblIncomeMapper;
    @Autowired
    private TblExpensesMapper tblExpensesMapper;
    @Autowired
    private TblInvtypeMapper tblInvtypeMapper;

    @Transactional
    @Override
    public int addUser(TblUser user) {
        TblUser user1 = tblUserMapper.queryUserByName(user.getUserName());
        if(user1==null){
            try{
                int effectedNum = tblUserMapper.insertUser(user);

                if(effectedNum>0){
                    TblUser u = tblUserMapper.queryUserByName(user.getUserName());
                    return u.getUserId();
                }else{
                    throw new RuntimeException("插入用户信息失败");
                }
            }catch(Exception e){
                throw new RuntimeException("插入用户信息失败:"+e.getMessage());
            }
        }
    }
}

```

TblUserController.java

```

@RestController
@RequestMapping("/user")
public class TblUserController{

    @Autowired
    private ITblUserService iTblUserService;

    @RequestMapping(value = "/addUser", method = RequestMethod.POST)
    private Map<String, Object> addUser(@RequestBody TblUser user){
        Map<String, Object> modelMap = new HashMap<>();
        Integer userId = iTblUserService.addUser(user);
        modelMap.put("userId",userId);
        return modelMap;
    }

    @RequestMapping(value = "/sumExcepses", method = RequestMethod.GET)
    private Map<String, Object> sumExcepses(Integer userId){
        Map<String, Object> modelMap = new HashMap<>();
        Integer sumExcepses = iTblUserService.sumExcepses(userId);
        modelMap.put("sumExcepses", sumExcepses);
        return modelMap;
    }

    @RequestMapping(value = "/sumInv", method = RequestMethod.GET)
    private Map<String, Object> sumInv(Integer userId){
        Map<String, Object> modelMap = new HashMap<>();
        Integer sumInv = iTblUserService.sumInv(userId);
        modelMap.put("sumInv", sumInv);
        return modelMap;
    }
}

```

TblUserMapper.xml

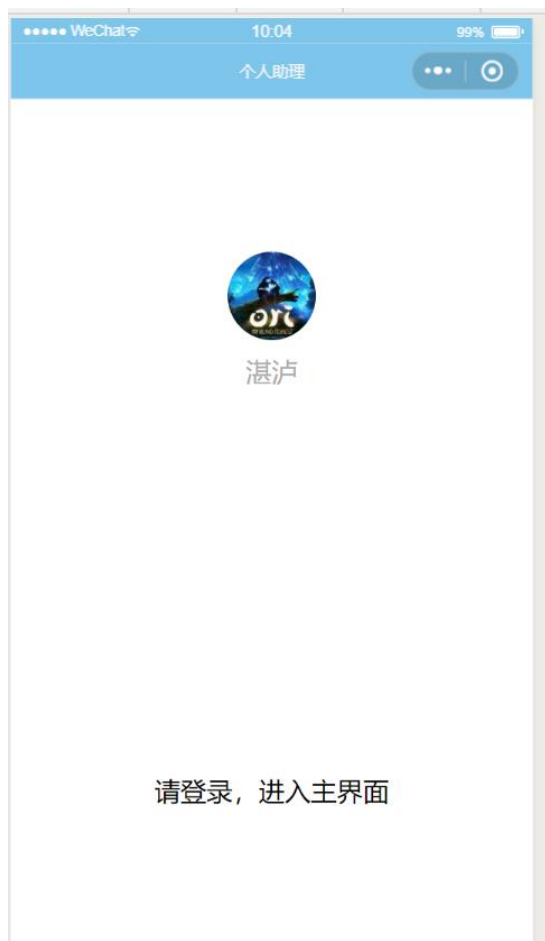
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org/DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.example.demo.dao.TblUserMapper">
  <select id="queryUserById" resultType="com.example.demo.entity.TblUser">
    SELECT *
    FROM tbl_user
    WHERE user_id=#{userId}
  </select>
  <select id="queryUserByName" resultType="com.example.demo.entity.TblUser">
    SELECT *
    FROM tbl_user
    WHERE user_name=#{userName}
  </select>
  <insert id="insertUser" useGeneratedKeys="true" keyProperty="userId"
    keyColumn="user_id" parameterType="com.example.demo.entity.TblUser">
    INSERT INTO
    tbl_user(user_name)
    VALUES
    (#{userName})
  </insert>
  <update id="updateUser" parameterType="com.example.demo.entity.TblUser">
    UPDATE tbl_user
    <set>
      <if test="userName!=null">user_name=#{userName}, </if>
      <if test="expCount!=null">exp_count=#{expCount}, </if>
      <if test="incCount!=null">inc_count=#{incCount}, </if>
      <if test="invCount!=null">inv_count=#{invCount}, </if>
      <if test="invPfcount!=null">inv_pfcoun=#{invPfcount}, </if>
    </set>
  </update>
</mapper>
```

5) 系统测试和展示

调试

展示

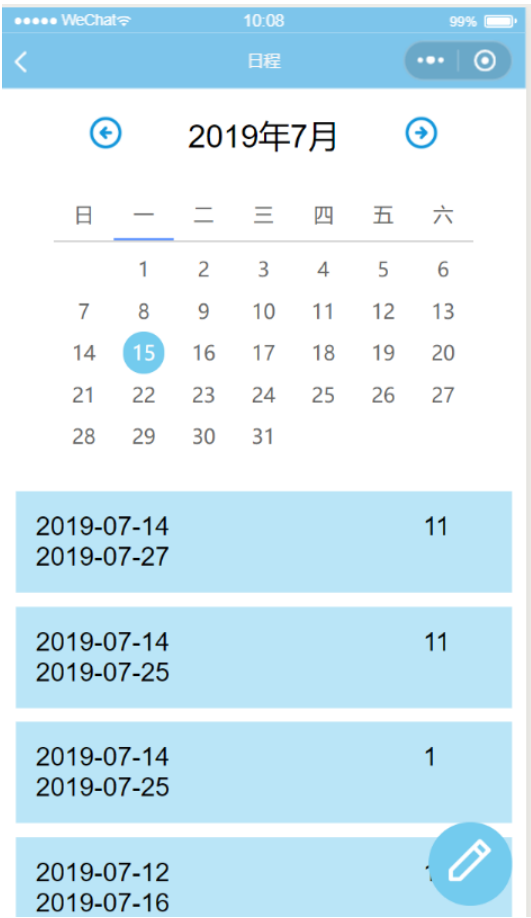
1. 点击头像后进入主界面



2. 显示主页及菜单



3. 显示当天日程及提醒



4. 点击右下角按钮添加日程



5. 长按所选提醒删除



6. 点击日记可查看日记



7. 点击右下角添加日记



8. 长按图片删除本条日记



9. 显示收入或支出列表



10. 点击右下角添加收支记录



11. 长按删除记录



12. 显示投资列表



13. 添加投资

..... WeChat

10:15

99%

<

增加投资

...

类型:

金额:

开始时间:

选择时间

利率:

提交

14. 按分类显示投资收益和金额

..... WeChat

10:15

99%

<

投资管理

...

⦿

投资列表

投资统计

总收益:

10

类型	金额	收益
	666	0
123	5000	250

+



三、收获

微信小程序在本短学期之前没有接触过，想趁此机会熟悉一下，因为很多地方会有这个需求，毕竟微信作为现在非常重要的一款社交软件，几乎人人使用，所以其中的微信小程序也有着一定的作用和地位。微信小程序与 APP 有较大的区别，更像是简化版的 VUE，有很多功能减少了，同时增加了其原生的功能。微信的接口和抓取数据没有像 APP 这么自由，APP 不需要服务器的域名的申请，微信只有部分的接口可以使用。要实现像消息推送，或者闹铃等这样的功能都是十分麻烦，有些功能只对大型企业级的用户开放（比如获取用户手机号）。消息推送也需要向微信平台进行申请，获取用户唯一标识 id 也需要向微信平台申请，所以微信小程序实际上有一定的局限性，这是要依附于微信平台存在的代价所在。

在写接口的时候，学会了前后端的联调，以及通过控制台查看请求的详细信息，成功失败的信息都会有，以及传输的数据包也会有。学会调试非常重要，可以及时的发现并改正问题，否则就会浪费大量时间在找错误上。微信小程序有利有弊，是值得学习和开发的，但同时 APP 的存在也无可替代。