****

**J I A N G S U U N I V E R S I T Y**

**“物联网移动应用开发”课程案例实践报告**

**题目：倾心家教及畜牧环控鸿蒙应用开发**

|  |  |
| --- | --- |
| 学院名称： | 计算机科学与通信工程学院 |
| 专业班级： | 物联网1903 |
| 学生姓名： | 杨靖宇 阮忠意 |
| 学生学号： | 3190611072 3190611071 |
| 指导教师： | 金 华 |

**2021年 11 月 24 日**

目录

[案例一：倾心家教 1](#_Toc30007)

[1.1案例简介 1](#_Toc20178)

[1.2需求分析 1](#_Toc6382)

[1.3项目设计 2](#_Toc22783)

[1.3.1数据库设计 2](#_Toc5138)

[1.3.2项目架构设计 5](#_Toc11699)

[1.3.3界面设计 6](#_Toc2612)

[1.4实现与测试 13](#_Toc6840)

[1.4.1 HarmonyOS客户端实现 13](#_Toc20614)

[1.4.2 SpringBoot服务端实现 18](#_Toc11457)

[1.5总结 21](#_Toc17426)

[案例二：面向畜牧养殖环境控制应用 23](#_Toc19271)

[2.1案例简介 23](#_Toc6814)

[2.2需求分析 23](#_Toc22627)

[2.3项目设计 24](#_Toc19364)

[2.3.1数据库设计 24](#_Toc7269)

[2.3.2项目架构设计 26](#_Toc23042)

[2.3.3界面设计 27](#_Toc17210)

[2.4实现与测试 31](#_Toc19545)

[2.4.1登录功能连接数据库 31](#_Toc8665)

[2.5总结 32](#_Toc20785)

# 案例一：倾心家教

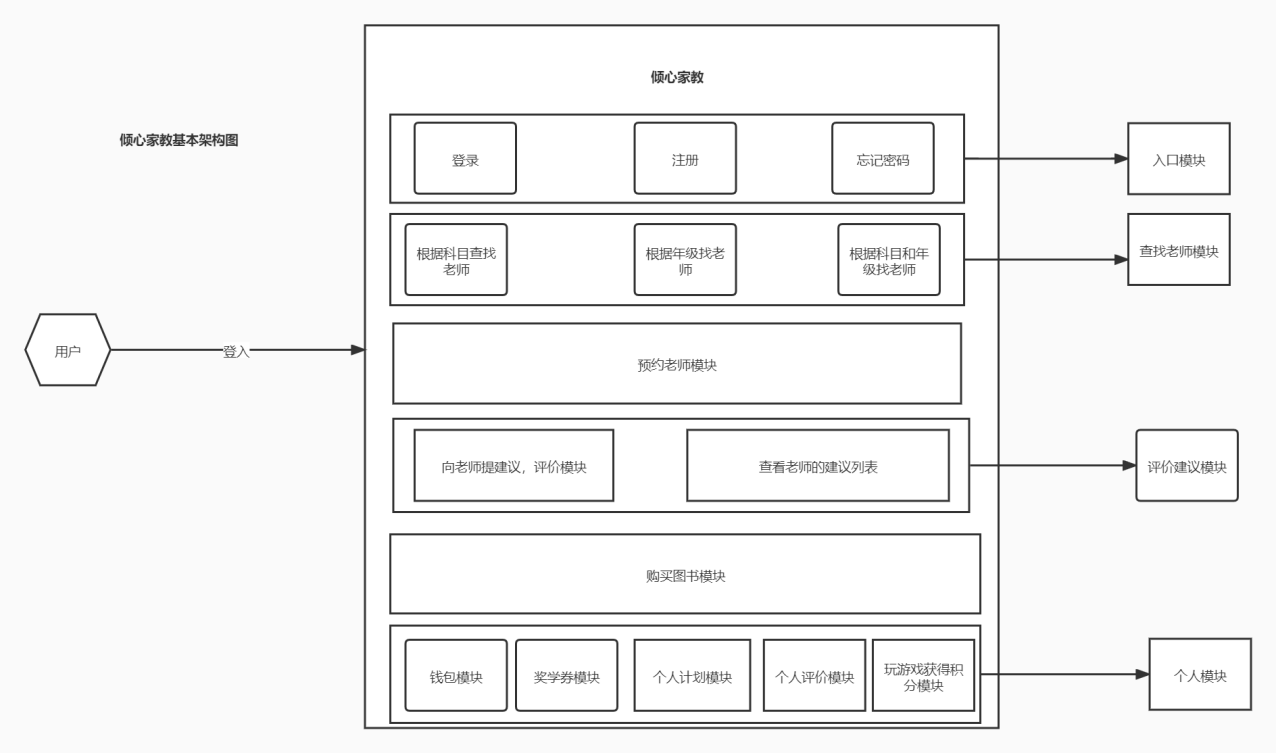
## 1.1案例简介

近年来,由于传统的培训机构成本飞涨,使得传统线下教育面临巨大压力;而且传统培训机构模式存在着大量的问题,比如时间、地点上的限制,师资力量问题,不能满足个性化的需求,等等。与之相反的是,在线查找教师和预约教师有很强的可选择性,可以满足不同用户的不同需要,随时随地,想学就学。极大地增加了用户的可选择度,降低了教育所带来的成本。

为了解决上述困境，我们推出了一款鸿蒙在线家教平台app——倾心家教，这款app旨在帮助所有学生突破因为教育资源匮乏而面临的困境。

这款app具有丰富的功能，不仅拥有查找老师，预约老师功能，还可以在app上购买相应的教辅资料。不仅如此，上完老师的课可以对老师提出有建设性的建议，还可以安排自己每天的时间做出当天计划等等功能。

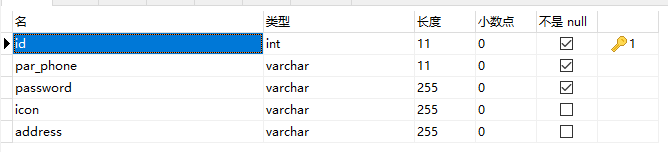
## 1.2需求分析

* 用户需要根据各种条件准确的查询到老师，点开老师可以看到老师的具体信息，包括老师的个人信息和评价以及他的可预约时间段，如果对该老师满意就可以进行预约，对已预约的老师也可以进行取消预约。
* 用户也可能需要教学配套的书籍，书籍可以进行购买与退换，另外购买书籍的同时可以用奖学券来抵消部分价格。
* 用户可以对教师进行中肯的评价，同时用户可以看到老师所有的被评价条目以供参考，另外也可以看到自己对所有老师的全部评价。
* 用户还可以查看个人计划，并且可以制定或者删除个人计划来规划自己的学习时间。
* 用户拥有自己私有的个人空间。他可以查看自己的钱包(余额，积分)，余额可以用来购买书籍，积分可以兑换奖学券。并且可以充值余额，积分可以通过玩小游戏获取。在个人空间可以查询到自己所有的信息，包括图书订单、教师预约、个人建议、个人计划等。

## 1.3项目设计

### 1.3.1数据库设计

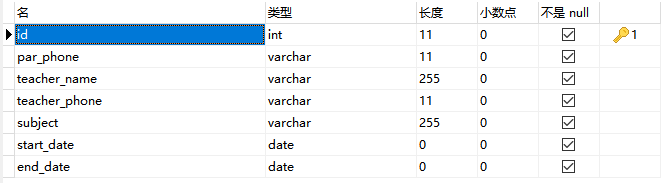
#### 1.3.1.1 parent表(用户表)

* Id：主键
* par\_phone：用户手机号
* password：用户登录密码
* icon：用户头像地址
* address：用户详细住址

#### 1.3.1.2 teacher表(教师表)

* id：主键id
* teacher\_name：老师姓名
* teacher\_sex：老师性别
* teacher\_phone：老师手机号码
* teacher\_address：老师详细地址
* teacher\_icon：老师头像地址
* teacher\_exper：老师教学经历
* subject：老师所教科目
* grade：老师所教年级

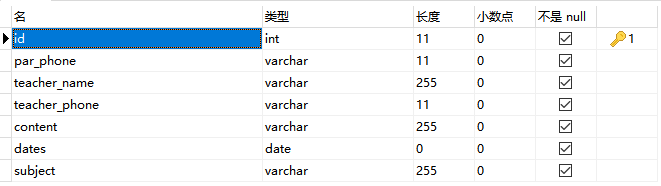
#### 1.3.1.3 reserve表(预约表)

* id:主键id
* par\_phone：用户手机号码
* teacher\_name:老师姓名
* teacher\_phone：老师手机号码
* subject：预约科目
* start\_date：预约开始时间
* end\_date:预约结束时间

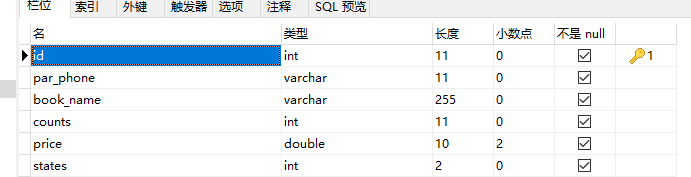
#### 1.3.1.4 plan表(计划表)

* id:主键id
* par\_phone：用户手机号码
* plan\_content:计划内容
* start\_date：计划开始时间
* end\_date:计划结束时间

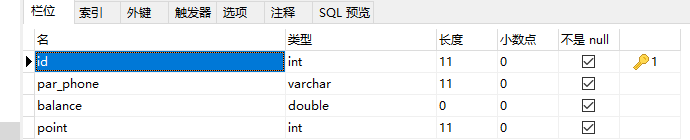
#### 1.3.1.5 evaluate表(评价表)

* id:主键id
* par\_phone：用户手机号码
* teacher\_name：老师姓名
* teacher\_phone：老师手机号码
* content：评价内容
* dates：评价日期
* subject：评价科目

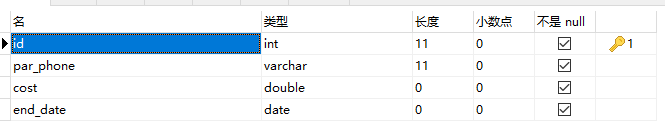
#### 1.3.1.6 book\_order表(订书表)

* id：主键id
* par\_phone：用户手机号码
* book\_name:购买的书籍名称
* counts：购买的数量
* price：所需的费用
* states：订单状态（0表示未发货，1表示已经收到书籍）

#### 1.3.1.7 par\_money表(钱包表)

* id:主键id
* par\_phone：用户手机号码
* balance：余额
* point：积分

#### 1.3.1.8 par\_reward表(奖学券表)

* id:主键id
* par\_phone：用户手机号码
* cost：奖学券面额
* end\_date：奖学券截止日期

### 1.3.2项目架构设计

倾心家教采用C/S架构，分为服务端与客户端。服务端部署在阿里云服务器上，因此客户端可以进行远程访问。

#### 1.3.2.1客户端

客户端为手机鸿蒙app，它与用户直接进行交互，采用xml+java进行编程。

#### 1.3.2.2服务端

服务端是一个RESTFul风格的SpringBoot整合MyBatis项目，它负责与MySql数据库进行交互，并向客户端提供相应的服务接口。

#### 1.3.2.3通信协议

服务端与客户端采用HTTP协议进行通信，客户端通过HTTP协议访问服务端相应接口获得数据再进行后续操作。

### 1.3.3界面设计

#### 1.3.3.1登录、注册、忘记密码页面设计



登录页面：

* 入口页面为登录页面，点击“登录”：登陆成功则跳转至主页面，失败则提示具体信息。
* 点击“注册”跳转至注册页面。
* 点击忘记密码跳转至忘记密码页面。

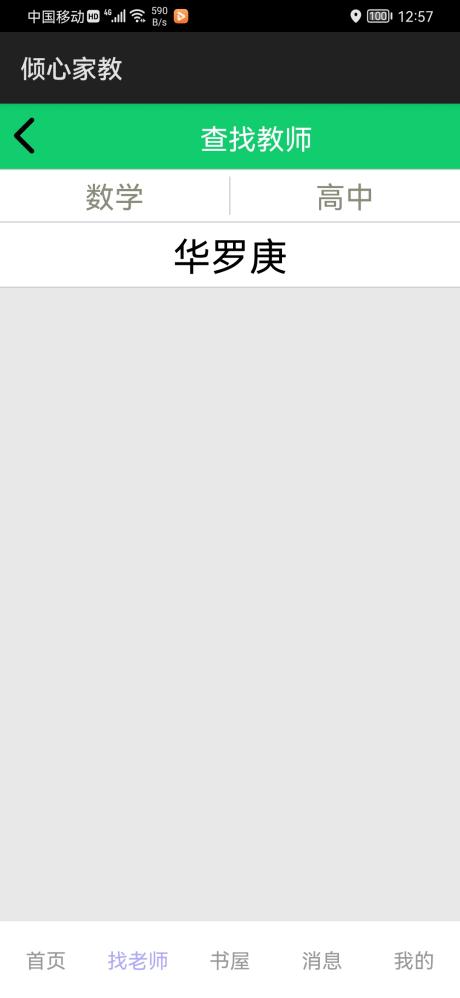
忘记密码页面：

* 点击“获取验证码”获取6位随机字符打印在按钮上。
* 点击“更改密码”，成功则跳回登录页面，失败则打印提示信息

注册页面：

* 点击“获取验证码”获取6位随机字符打印在按钮上。
* 点击“注册”，会有Toast提示信息。
* 点击“返回登录”跳转至登录页面。

#### 1.3.3.2主页面设计



* 首页最左上角显示“镇江市”：自动获取地理位置，前提是授予app获取地理位置的权限。
* 首页中上部分有8门课程，点击可以跳转至找老师页面，并且是带条件跳转，找老师页面只会显示该门课程的老师
* 首页中间本分是一个滑动的广告窗口，用于显示各种广告。
* 首页中下部则是热门名师窗口，展示本平台最热门的几位老师，点击可以进去该老师详细信息界面。
* 最下部分则是一个导航栏，用于主页面之间的跳转。
* 找老师界面点击科目会跳转到选择科目界面，并带着结果返回该界面，同理年级按钮



#### 1.3.3.3倾心书屋页面设计



* 倾心书屋界面提供9本教辅资料供用户选择，点击心仪教辅则跳转至第二个页面。
* 第二个页面提供了该书籍的单价，输入购买数量点击提交跳转至第三个页面。
* 第三个页面显示了当前订单的总价，并提供抵用券（奖学券）的使用，点击提交订单则进行相应的扣款处理并提交订单。

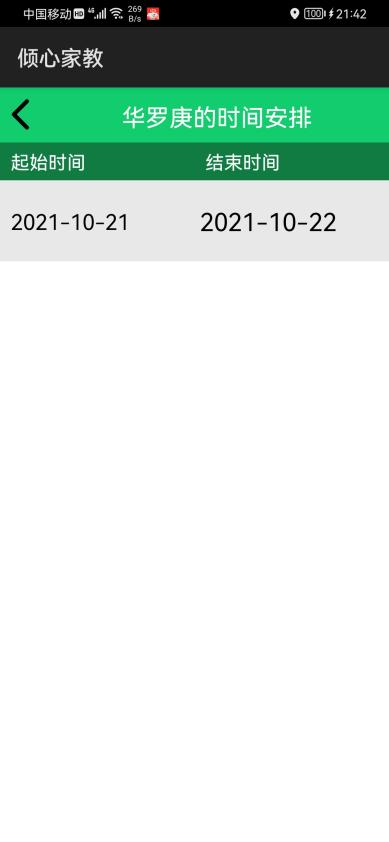
#### 1.3.3.4 消息页面设计



* 在消息通知页面中，点击预约老师信息就会跳转到第二张图，点击我要评价就会跳转到第三张图
* 第三张图中点击科目会跳转至另一个页面选择科目然后带着结果返回，同理教师，填写好评价后点击提交即可提交成功。

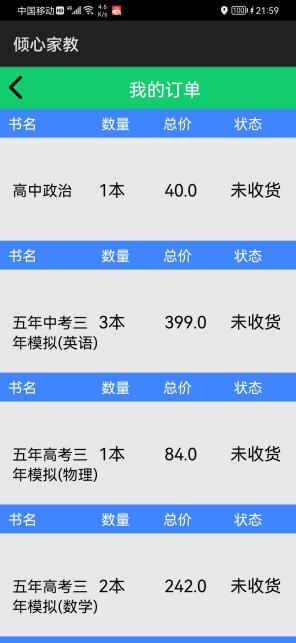
#### 1.3.3.5预约老师页面设计

选取一个老师点击进入后会显示当前老师的详细信息（第一张图）



* 第一张图显示当前老师的详细信息，点击时间安排表会跳转至第二张图；点击预约老师则会跳转至第三张图
* 第二张图会显示当前老师已被预约的时间。提醒用户不要预约冲突
* 第三张图填写预约表然后提交即可

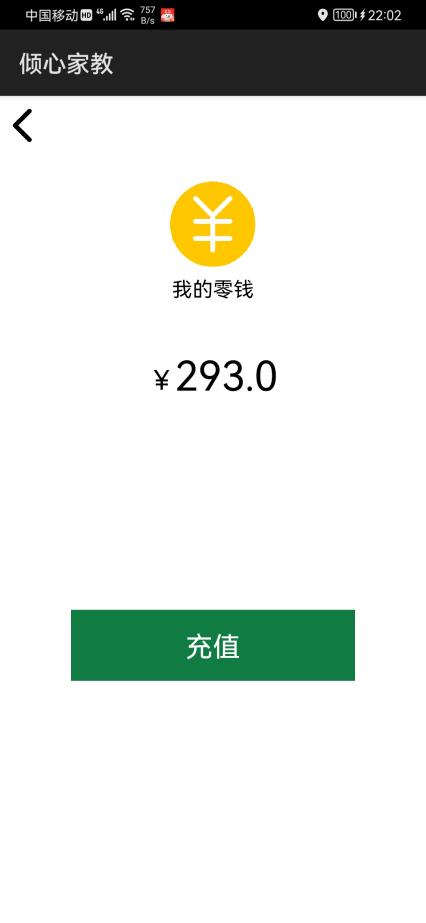
#### 1.3.3.6个人信息页面设计



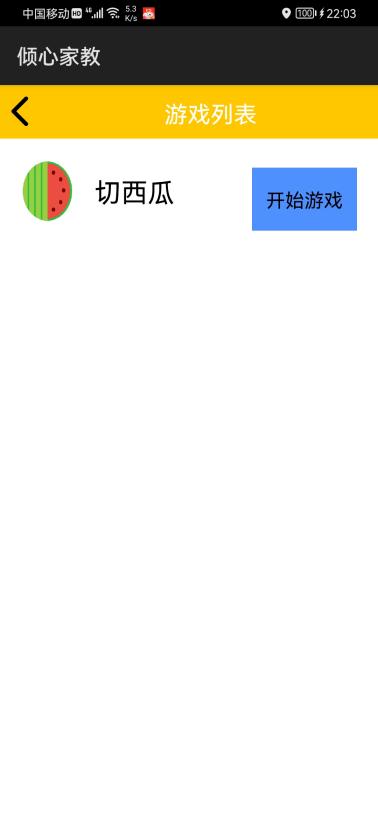
* 在个人信息页面中点击“我的订单”则会跳转至第二张图，点击“我的钱包”则会跳转至第三张图，点击“奖学券”则会跳转至第四张图。
* 在我的订单中点击任意条目则会跳转至右图，点击删除订单即可删除

相应订单。

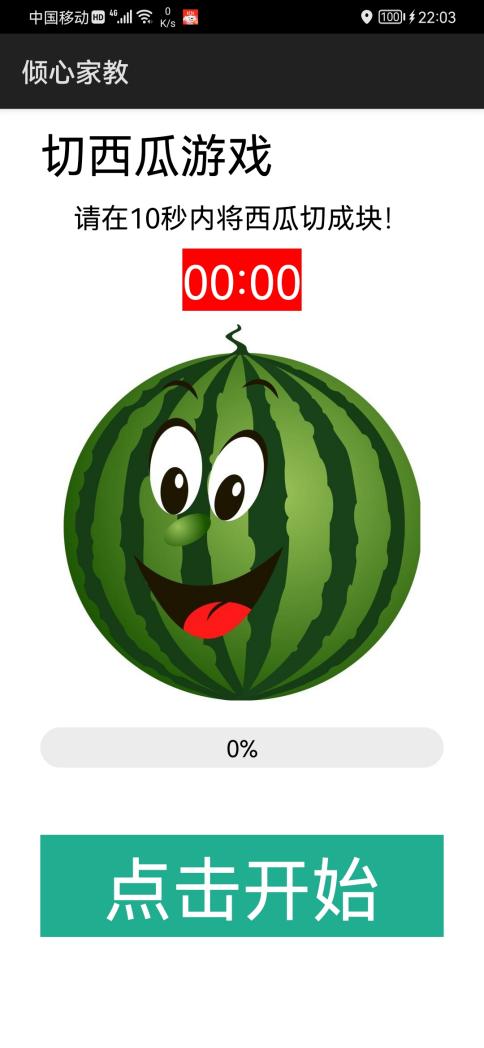
* 在我的钱包中，点击我的余额会跳转至如下页面：

点击充值跳转至第二张图。

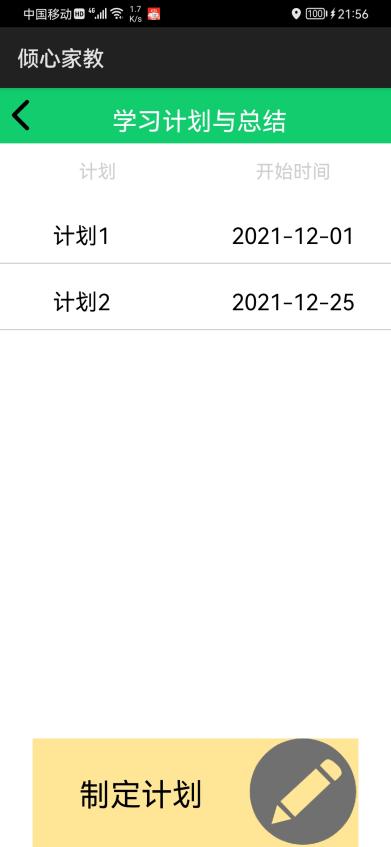
* 在我的钱包中，点击积分则会跳转至如下页面：



* 点击兑换积分则会跳转至第二个页面，在第二个页面中点击具体奖学券即可进行兑换。
* 点击“获取积分”则会跳转至第三个页面，玩游戏获取积分：



* 在个人界面中，点击“我的老师”,“我的评价”，“学习计划与总结”：



* 点击“我的老师”和“我的评价”页面中具体的某一项时会跳转显示具体信息，原理上文已提到，此处不再赘述。
* 在学习计划与总结中，点击制定计划跳转至第二张图



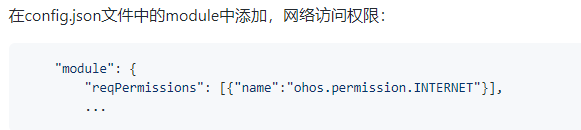
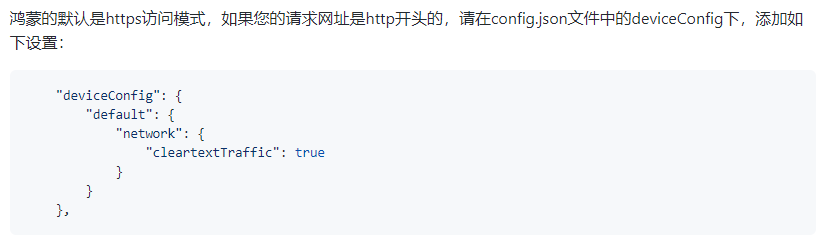
* 设置中点击退出账号：



## 1.4实现与测试

### 1.4.1 HarmonyOS客户端实现

#### 1.4.1.1 HTTP框架实现与具体配置

* 导入依赖
* 授予权限
* 修改默认访问配置
* 常用ZZRHTTP框架的方法
* postJson(String url, String jsonStr, ZZRCallBack callBack)
* get(String url, ZZRCallBack callBack)

#### 1.4.1.2 Gson框架的使用

* 创建Gson对象

Gson gson= new GsonBuilder().serializeNulls().setDateFormat("yyyy-MM-dd").create();

* 将实体类转化为json

String userJson = gson.toJson(user)

* 将json转化为实体

Parent user = gson.fromJson(userJson,Parent.class)

* 将json转化成装实体的List

List<Parent> list = gson.fromJson(jsonList, new TypeToken<List<Parent>>() {}.getType());

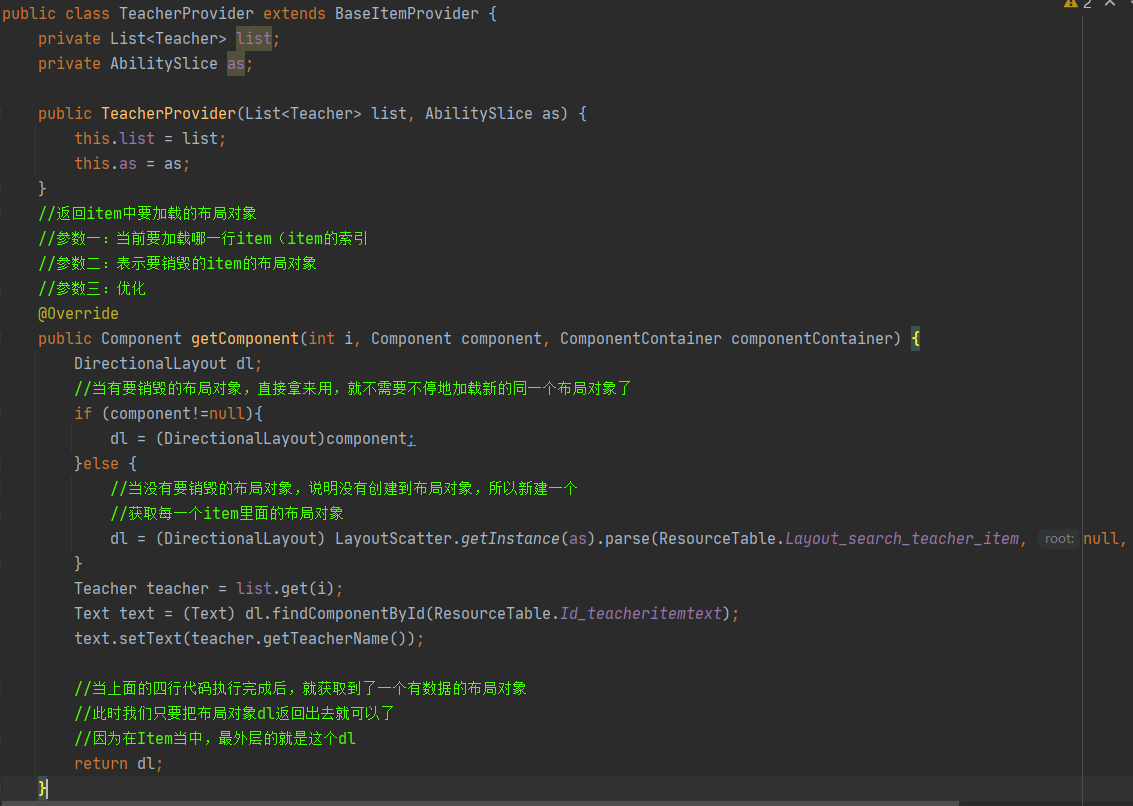
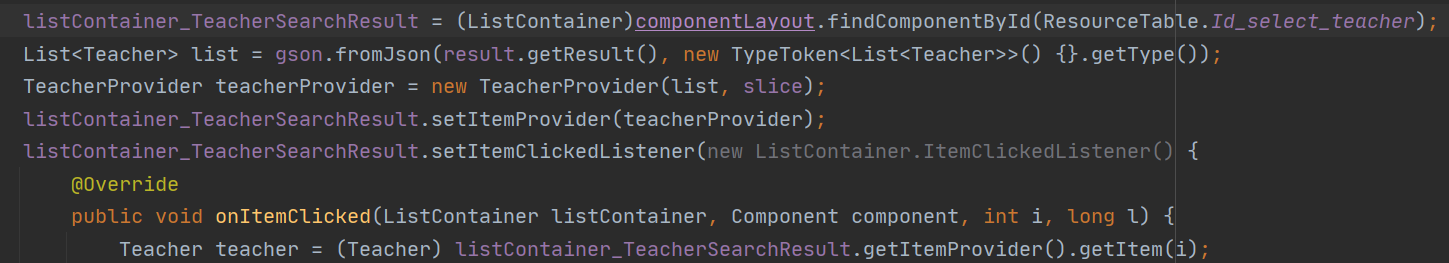
#### 1.4.1.3 登录示例（ZZRHttp整合Gson使用示例）



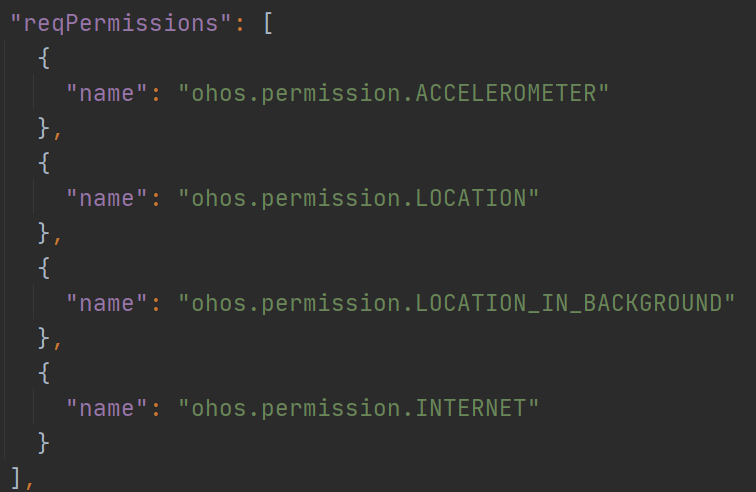
#### 1.4.1.4 导航栏的实现

* 使用tablist和tab组件在页面底部放置标签实现导航栏效果
* 使用回调方法onSelected实现导航栏对应的布局切换效果

#### 1.4.1.5 ListContainer+item动态展示数据列表列表

* 创建item和item适配器，适配器将item中要展示的信息设置好
* 将适配器与list container组件关联起来，就可以在list container组件中展示item信息

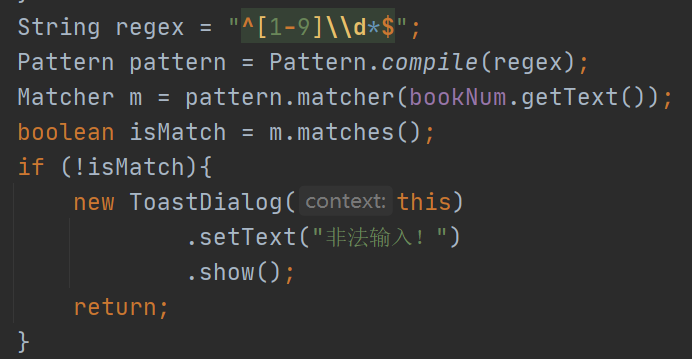
#### 1.4.1.6 地理位置的获取

* 在config.json中声明获取地址和连接网络的动态权限
* 通过Locator及其管家获取到当前经纬度坐标，并且连接网络将经纬度坐标转换成对应的城市名字。

#### 1.4.1.7 随机验证码的获取

* 通过工具类来随机生成任意长度的验证码。

#### 1.4.1.8 正则表达式的使用



* 通过正则表达式快速检验用户输入的字符串是否合法，提高程序的容错率。

### 1.4.2 SpringBoot服务端实现

#### 1.4.2.1 Result类（HTTP协议中的数据包）

* 当操作成功时调用setSuccess（）方法
* 当操作失败时则调用setInfo（）方法

#### 1.4.2.2 实体类

* 为了体现ORM思想，实体类与数据库一一对应，此处不再赘述。

#### 1.4.2.3 application.properties配置文件

#### 1.4.2.4 Controller层

1. ParentController实现的接口

* /login（登录）
* /register（注册）
* /forgetPassword（修改密码）

1. TeacherController实现的接口

* /teacherList（所有老师）
* /findTeacherByName（通过老师姓名找老师）
* /findTeacherBySubject（通过课程找老师）
* /findTeacherByGrade（通过年级找老师）
* /findTeacherBySubjectAndGrade（通过课程与年级找老师）
* /findTeacherByPhone（通过老师手机号码找老师）

1. ReserveController实现的接口

* /makeReserve（预约老师）
* /reserveList（老师的预约列表）
* /deleteReserve（取消预约）
* /myReserveList（我的预约列表）

1. PlanController实现的接口

* /todayPlan（我的今天的计划）
* /myPlan（我的计划）
* /makePlan（新建计划）
* /cleanPlan（清理7天之前的计划）
* /deletePlan（删除计划）

1. EvaluateController实现的接口

* /teacherEvaluateList（老师的评价列表）
* /myEvaluateList（我的评价列表）
* /makeEvaluate（提出评价）
* /deleteEvaluate（删除评价）

1. BookOrderController实现的接口

* /myOrderList（我的订单）
* /orderBook（购买书籍）
* /cancelOrder（取消订单）

1. ParMoneyController实现的接口

* /myMoney（我的钱包）
* /recharge（充值/消费）

1. ParRewardController实现的接口

* /myReward（我的奖学券）
* /giveReward（获得奖学券）
* /useReward（使用奖学券）

#### 1.4.2.5 Service层

业务层，一般进行业务的解耦。

#### 1.4.2.6 Mapper层

与数据库进行交互的一层，在这一层中写SQL语句。

#### 1.4.2.7 SpringBoot项目的部署运行

首先用package命令将工程打成jar包，然后上传到服务器，在相应的目录下输入：

* nohup java -jar xxx.jar &
* exit

## 1.5总结

**杨靖宇：**

完成倾心家教这个案例可以说是困难重重，耗费了我们大量的精力。由于鸿蒙系统刚出不久，网上除了官方文档什么都没有，而官方文档不仅晦涩难懂，而且也不能满足我们的开发需求，但还好网上有零散的一些鸿蒙开发教程，它对官方文档的基本解读让我们入了门。但是我们接下来开发鸿蒙版倾心家教就困难重重了。

在开发过程中，我们首先碰到的问题是导航栏的实现，在我的第一版中，导航栏是利用碎片化Slice，即Fraction。它的样式与Slice差不多，但最不好的一点就是他不能实现点击等各种事件的接口，这造成了所有点击事件必须写在导航栏那个类中，造成了那个类十分臃肿。后来我们查阅大量资料，才找到最优替换方法，上文已经提到，此处不再赘述。

第二个难点就是LstContainer+item 动态生成列表，视频教程中有但比较难以理解，另外还有就是地理位置的获取，这个教程中也有，但当时对我们来说还是比较难的。

最后也是最最困难的问题——数据库的连接，虽然鸿蒙提供SQLLit这种嵌入式数据库。但是也正是这种嵌入式数据库不可以可视化操作，联机操作也比较困难，简单的来说就是非常的不好用，因此我们想要用想Mysql/SQLServer这种关系型数据库，并且安卓版倾心家教也是提供了安卓+PHP+MySQL教程，按理说鸿蒙也没多大问题。但是多次尝试之后发现不可行，找到原因是鸿蒙是远程模拟器连不了本地数据库。除非内网穿透。于是我改连我阿里云的数据库，但此时又不行说耗时任务不能在主进程中运行，但是后来我用简单的任务分发器访问还是不行，至此我已经有放弃的念头了。后来逛GitHub偶然间发现一个鸿蒙网络框架ZZRHttp，突然有了灵感，于是就采用了C/S架构并取得了完美的成功。

总的来说，还是收获良多，我现在也可以说是一名鸿蒙应用开发工程师了，不过很不辛的是，鸿蒙正处于快速更新迭代中，估计要不了多久，项目中使用的技术就过时了。不得不说，这次案例开发真是体验感十足。

**阮忠意：**

本次倾心家教APP开发中，虽然有Android版本的案例pdf教程，但是这次开发使用的基于鸿蒙系统的DevEco Studio开发工具还是与Android版本存在一定差异，并且版本也仅是2.2.0，想要对着Android案例教程实现组件功能，DevEco Studio还是有一部分组件没有，官网的文档教程和黑马程序员教程里出现的组件都是常见的组件，当发现没有对应的组件可以实现想要实现的功能时，只能网上到处搜寻资料，查找同时正在学习开发鸿蒙系统app的相关博主分享的资源、第三方库等，才能学习到实现像导航栏、滚动图片的选择等功能。本次项目主要的两个难点：

首先，在实现导航栏的时候，在网上找到使用tablist和tab组件实现简单的导航栏功能资料后，花了一定时间了解tablist和tab组件和布局XML组件的关系，他有区别于使用DevEco Studio开发工具搭建普通的项目页面，用一个java ability中的ability slice对应一个要展示的页面XML文件，而本项目的导航栏实现是用一个slice对应的一个XML中放置tablist组件和direction layout组件，通过java代码在slice中将点击的tablist中的tab对应的要显示的页面XML文件，加载到direction layout组件中，即可实现导航栏的功能。本次项目在实现了导航栏中页面的切换后，还遇到了一个难题：想要找到每个通过导航栏点击出来的XML文件里的组件，需要设计一个代码的逻辑框架，将根据导航栏的标签点击展示的XML，找到这个XML布局对象，传入自定义的方法框架里，才可将该XML中的组件找到，就可以对组件进行功能设置。

其次，就是进行前后端的交互，将后端提供的接口，通过ZZRHTTPs类，连接到服务端，访问服务端的数据库，获得服务端的数据库信息，用来给组件实现对应的功能。

破解了这两个难题，剩下的就是将所有XML设计和所有相关功能实现的设计在DevEco Studio开发工具编写出来。此次项目在修改调试bug阶段也费了不少心思。

总的来说，为了实现用鸿蒙操作系统开发倾心家教的app，把网上的视频教程中前端的设计都看完了，查阅了各种相关前端功能设计的资料，同时也为了实现连接服务端的功能，也学习了一下基于鸿蒙网络框架ZZRHttp连接服务端的知识，最后通过实战的编写了这次数千条代码的项目，深刻掌握了使用鸿蒙系统开发app的前端设计部分，了解了连接服务端的部分知识。

# 案例二：面向畜牧养殖环境控制应用

## 2.1案例简介

最传统的行业往往最容易发生颠覆性变革。曾经被认为“又脏又累”的养殖行业，工业互联网的落地速度反而超乎想象。

在智慧养猪场，每一头猪从出生开始就进行身份认证，喂料、喂水、通风、保温、粪便处理都是全自动，场区还专门安装除臭设备对臭气进行“净化”。即便是在炎热的天气里，猪舍里装有负压风机和湿帘等设备，温度能精准控制在体感温度20℃左右，让猪保持最佳生长状态。

所有动态数据都通过养殖集团信息化智能分析管控系统进行采集、传输和分析，实时监控猪舍环境和生猪的生存状况，对养殖生产管理进行有效指导。管理员通过一台电脑、一部手机，就能实时掌握养殖场环境信息，及时获取异常报警信息，并远程控制相应设备。

“一个万头猪场只需要几个人管理，将来人员可能更少，既提高了生产效率，也降低了人员接触带来的疫病传播风险，保证养殖环境处于最优状态。”在大牧人智能猪场分析管控系统首席架构师赵世光看来，借助工业互联网技术，在可预见的未来，无人值守猪场一定可以实现。

物联网智能环境控制系统主要用于密闭式畜禽舍通风控制。根据温湿度传感器的信号，通过通风、加热、降温等措施，将畜禽舍的环境温湿度控制在一个设定的范围，以满足畜禽成长的要求。

## 2.2需求分析

功能具体如下：

1）触摸屏显示；

2）实时记录各个报警发生的日期和时间；

3）可接入 5 个采集节点（P5 可用于室外），每个采集节点包括温湿度传感器、CO2

浓度传感器以及 NH3 浓度传感器；

4）2 组变速风机控制、8 组全速风机控制；

5）2 组加热器；

6）1 组报警器；

7）温度曲线，最小通风曲线功能；

8）温度补偿、湿度补偿功能；

9）卷帘和通风窗控制；

10）可 4G 连接到物联网监控系统，随时随地进行数据浏览；

## 2.3项目设计

### 2.3.1数据库设计

#### 2.3.1.1、用户表user\_Data

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 主键 | **说明** |
| **username** | varchar(100) | √ | 用户名 |
| **userpassword** | varchar(100) |  | 密码 |

#### 2.3.1.2、采集节点数据表collect\_data

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 主键 | **说明** |
| **datetime** | varchar(20) | √ | 日期时间 |
| **collect\_num** | varchar(5) | √ | 采集节点编号 |
| **collect\_id** | varchar(10) |  | 采集节点id |
| **temp** | double |  | 温度 |
| **humi** | decimal |  | 湿度 |
| **co2** | decimal |  | Co2浓度 |
| **nh3** | decimal |  | Nh3浓度 |

#### 2.3.1.3、控制器数据表control\_data

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 主键 | **说明** |
| datetime | varchar(20) | √ | 日期时间 |
| control\_id | varchar(10) |  | 控制器id |
| light | decimal |  | 光照强度 |
| relay1 | decimal |  | 继电器的状态（1表示开，0表示关） |
| relay2 | decimal |  | 继电器的状态（1表示开，0表示关） |
| relay3 | decimal |  | 继电器的状态（1表示开，0表示关） |
| relay4 | decimal |  | 继电器的状态（1表示开，0表示关） |
| relay5 | decimal |  | 继电器的状态（1表示开，0表示关） |
| relay6 | decimal |  | 继电器的状态（1表示开，0表示关） |
| relay7 | decimal |  | 继电器的状态（1表示开，0表示关） |
| relay8 | decimal |  | 继电器的状态（1表示开，0表示关） |
| relay9 | decimal |  | 继电器的状态（1表示开，0表示关） |
| relay10 | decimal |  | 继电器的状态（1表示开，0表示关） |
| relay11 | decimal |  | 继电器的状态（1表示开，0表示关） |
| relay12 | decimal |  | 继电器的状态（1表示开，0表示关） |
| relay13 | decimal |  | 继电器的状态（1表示开，0表示关） |
| relay14 | decimal |  | 继电器的状态（1表示开，0表示关） |
| relay15 | decimal |  | 继电器的状态（1表示开，0表示关） |
| relay16 | decimal |  | 继电器的状态（1表示开，0表示关） |
| v1 | decimal |  | 模拟量输出的电压 |
| v2 | decimal |  | 模拟量输出的电压 |
| v3 | decimal |  | 模拟量输出的电压 |
| v4 | decimal |  | 模拟量输出的电压 |
| v5 | decimal |  | 模拟量输出的电压 |
| v6 | decimal |  | 模拟量输出的电压 |

#### 2.3.1.4、控制器信息表control\_info

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 主键 | **说明** |
| control\_id | varchar(10) | √ | 控制器id |
| user\_id | varchar(10) |  | 该控制器的买家id |
| panid | char(4) |  | 控制器ZigBee模块的PANID |
| channel | decimal |  | 控制器ZigBee模块的信道 |

#### 2.3.1.5、登录日志表login\_log

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 主键 | **说明** |
| datetime | varchar(20) | √ | 控制器上线日期 |
| control\_id | varchar(10) |  | 控制器id |

#### 2.3.1.6、控制器用户信息表user\_info

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 主键 | **说明** |
| user\_id | varchar(10) | √ | 用户id |
| username | varchar(20) |  | 用户名 |
| detail | varchar(100) |  | 用户详细信息 |

### 2.3.2项目架构设计

面向畜牧养殖环境控制应用采用C/S架构，分为服务端与客户端。服务端部署在阿里云服务器上，因此客户端可以进行远程访问。

#### 2.3.2.1客户端

客户端为手机鸿蒙app，它与用户直接进行交互，采用xml+java进行编程。

#### 2.3.2.2服务端

服务端为Django项目，它与数据库进行交互，并为客户端提供接口。

#### 2.3.2.3通信协议

服务端与客户端采用HTTP协议进行通信，客户端通过HTTP协议访问服务端相应接口获得数据再进行后续操作。

### 2.3.3界面设计

#### 2.3.3.1开机界面

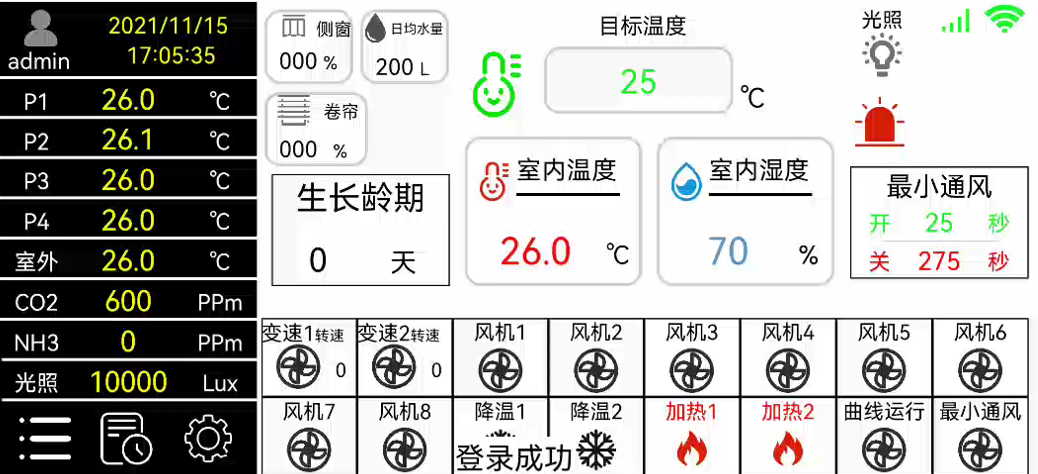


* 点击后跳转进入主页面

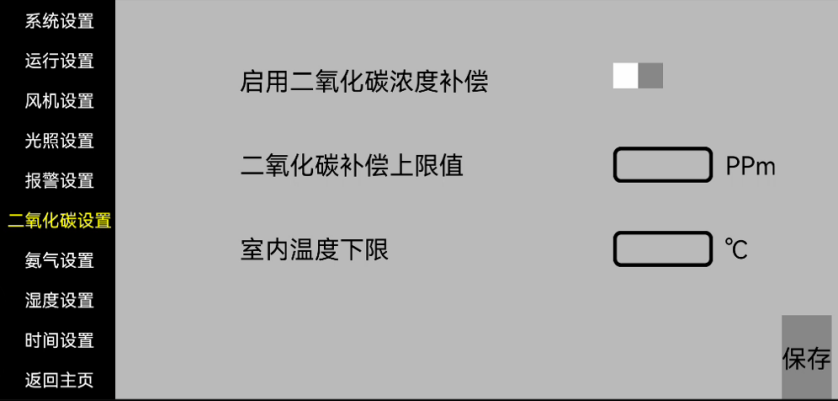
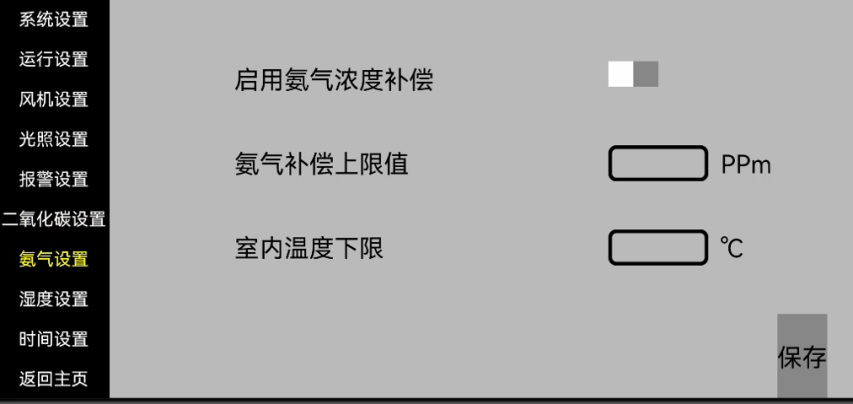
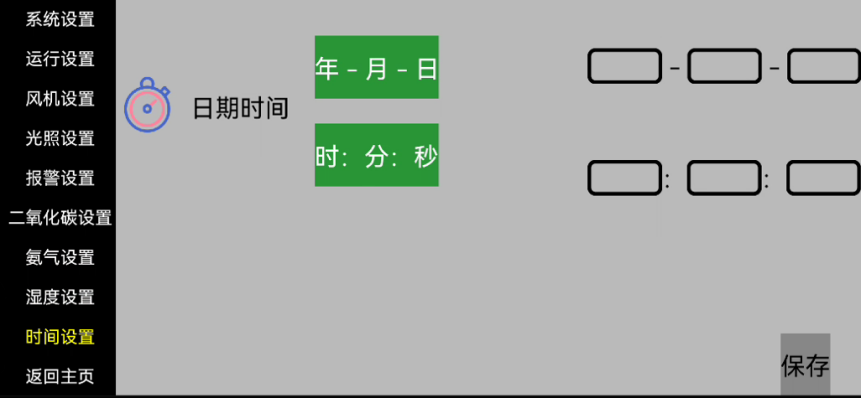
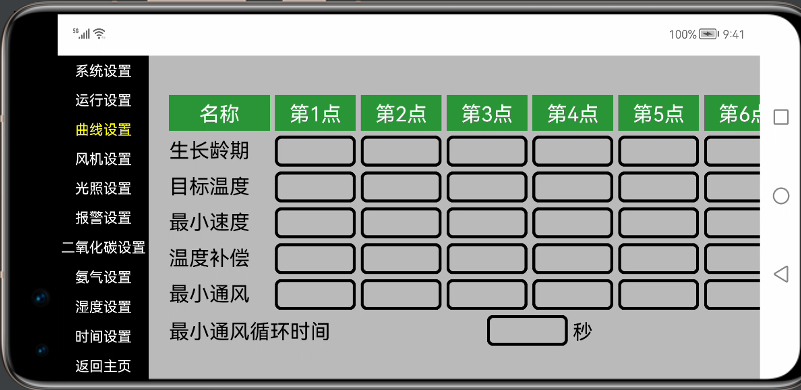
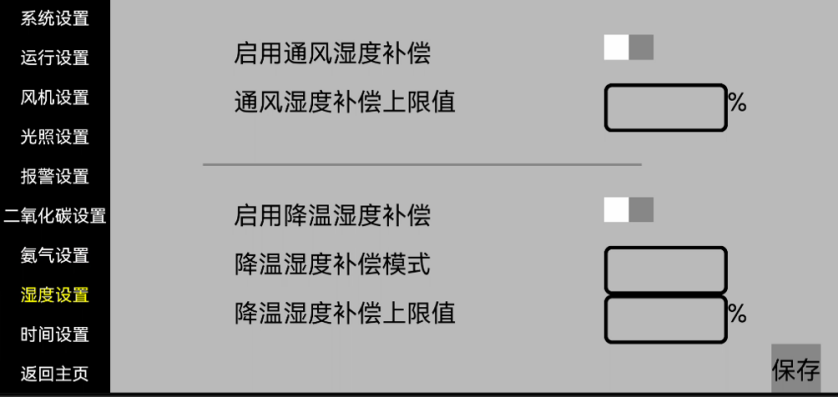
#### 2.3.3.2系统主页面

* 通过开机界面进入的系统主页面，初始化时用户是未登录状态

#### 2.3.3.3 用户登录和修改密码页面

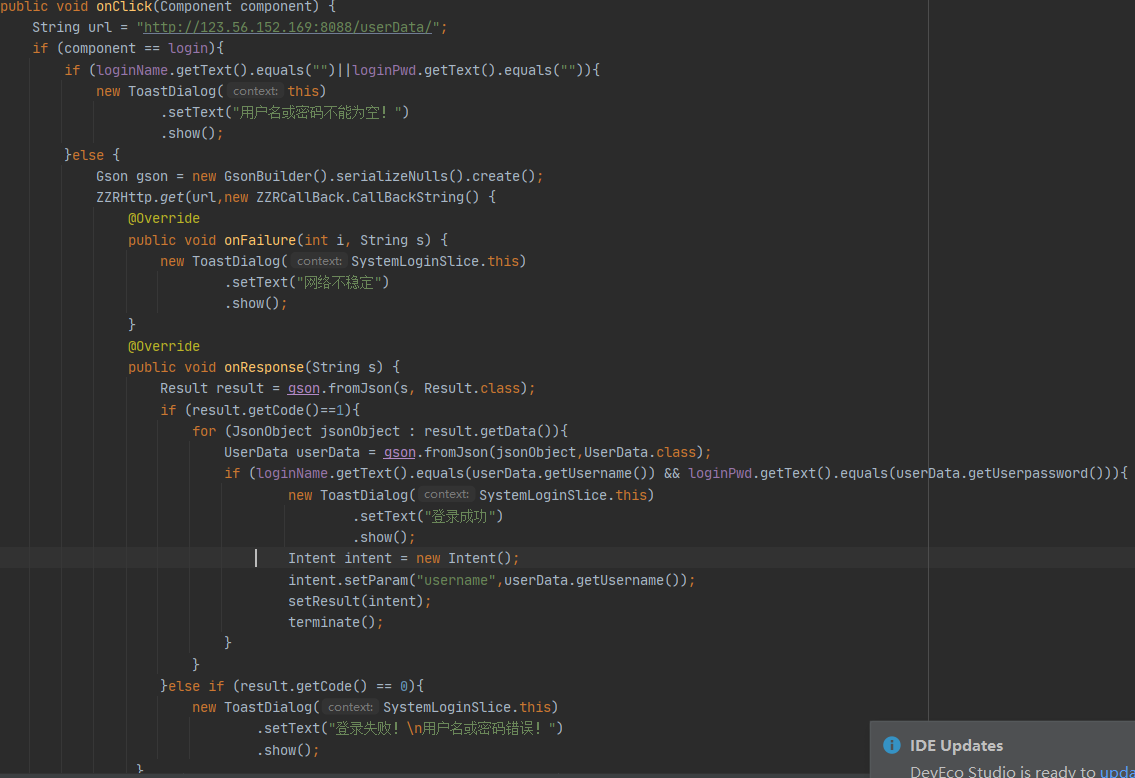
* 点击系统主页面的用户头像或登录状态即可进入用户登录页面
* 输入账号和密码后登录会通过对比数据库中保存的账号密码来判断是否登录成功
* 用户登录页面也可以进行密码修改

#### 2.3.3.4 系统设置页面



## 2.4实现与测试

### 2.4.1登录功能连接数据库

* 创建Result类，接收连接数据库返回的包。
* 根据给出的数据库接口，根据数据库返回的账号密码判断输入的账号密码正确性

**剩余模块功能与倾心家教类似，此处不再详细说明。**

## 2.5总结

**杨靖宇：**

本案例有了上一个案例作为基础，写起来简直就是行云流水，本质上和上一个应用没有太大区别，因此许多模式可以照搬倾心家教。

我觉得本案例最主要的难点在于概念的理解，养殖场有许多参数，记住每个参数以及它的意义是比较困难的，因为我们对于养殖场并没有什么概念，这也是最难点所在。不像倾心家教十分贴近生活，我们对那些概念十分熟悉，理解起来没有难度。

总的来说，就当是一个小倾心家教练手项目了，又加深了一次对鸿蒙的理解。

**阮忠意：**

本次案例主要的任务就在前端的页面设计，基于案例1掌握到的前端设计基础，做案例2的页面就没有困难了，花费的就是设计出十几个页面用XML文件敲代码的体力和时间，数据库提供的接口也少，连接数据库实现功能的模块也简单，案例2的项目很容易就做完

了。