



校易

系统设计和数据库设计

码到成功小组制作



系统设计部分

- 项目背景
- 系统总体结构
- 系统设计
- 数据库设计
- 界面设计

A



项目背景

随着现在诸多高校的扩招与校园信息化的开展，校园网络的建设将进入一个更加高速发展的时期。这对于我们来说也将有一个更好的环境去发展校园电子商务，建立属于高校大学生自己的交易平台。

当今社会大学生的人口基数已不少于任何一群体，高校学生同时也是一个高消耗人群，对于商品资源的使用自然不会少，要做到资源充分利用，同时也响应党十七大“节能减排”的号召，立足于满足高校学生二手交易的需要，二手交易平台显得更加必不可少。同为大学生，大家都具有更相似的消费心理，共同的生活环境，生活方式，所以之间交易的商品也都是各自需要的，从这一角度来看，二手交易平台的构建更具可行性和必要性。



功能设计

1. 用户登录：用户注册、学号验证
2. 首页：搜索、我要买、我要卖
3. 商品详情：商品信息、私聊界面
4. 用户查看卖家：卖家信息
5. 个人中心：我的商品、我的商品出售中、我的商品已下架
6. 管理员：下架用户商品、注销用户账号



实现约束

- 人员配合：假设在项目开发过程中，全队队员能按时高效地完成队长分配的任务，在推广过程中能得到支持。
- 软件开发平台支持：假设在软件开发中所涉及的开发工具和平台能很好地支持开发，例如：在weui上可以很好地完成可视化界面的塑造。
- 资金限制：假设项目有足够的启动资金。
- 时间限制：假设项目完成时间不会推移。
- 可操作性：假设大部分用户对于本产品都能灵活操作。
- 用户配合：假设向用户推广此产品时，大家都能够接受以及认可本产品。



系统总体结构

1.技术架构：小程序客户端采用C/S（小程序客户端—服务器）架构，管理端采用B/S架构（浏览器—服务器）。

使用PHP三层体系架构：

2.表示层：主要使用WEB-Render方式，逻辑层强大和完善，无论表现层如何定义和更改，各司其职，逻辑层都能完善地提供服务。

3.业务逻辑层：主要针对具体问题操作，也是对数据Data层的操作，对数据业务进行逻辑处理，实现积木拼接式搭建。

抽象接口层：对数据访问层抽象出接口，业务逻辑层经过抽象接口层去调用，保证调用分离，扩展分离。

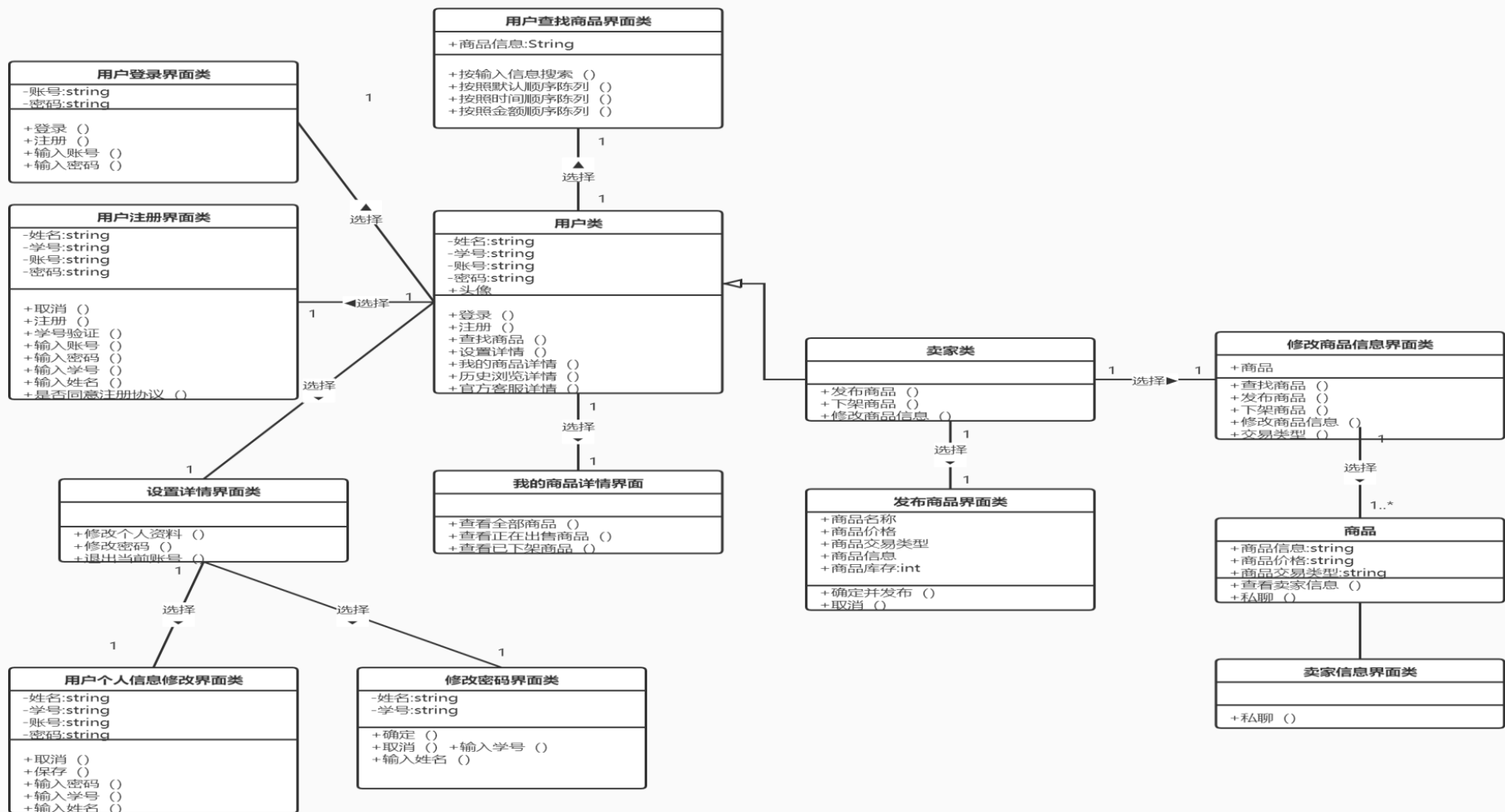
4.数据访问层：主要对原始数据进行加工和提取，为业务逻辑层提供数据服务。



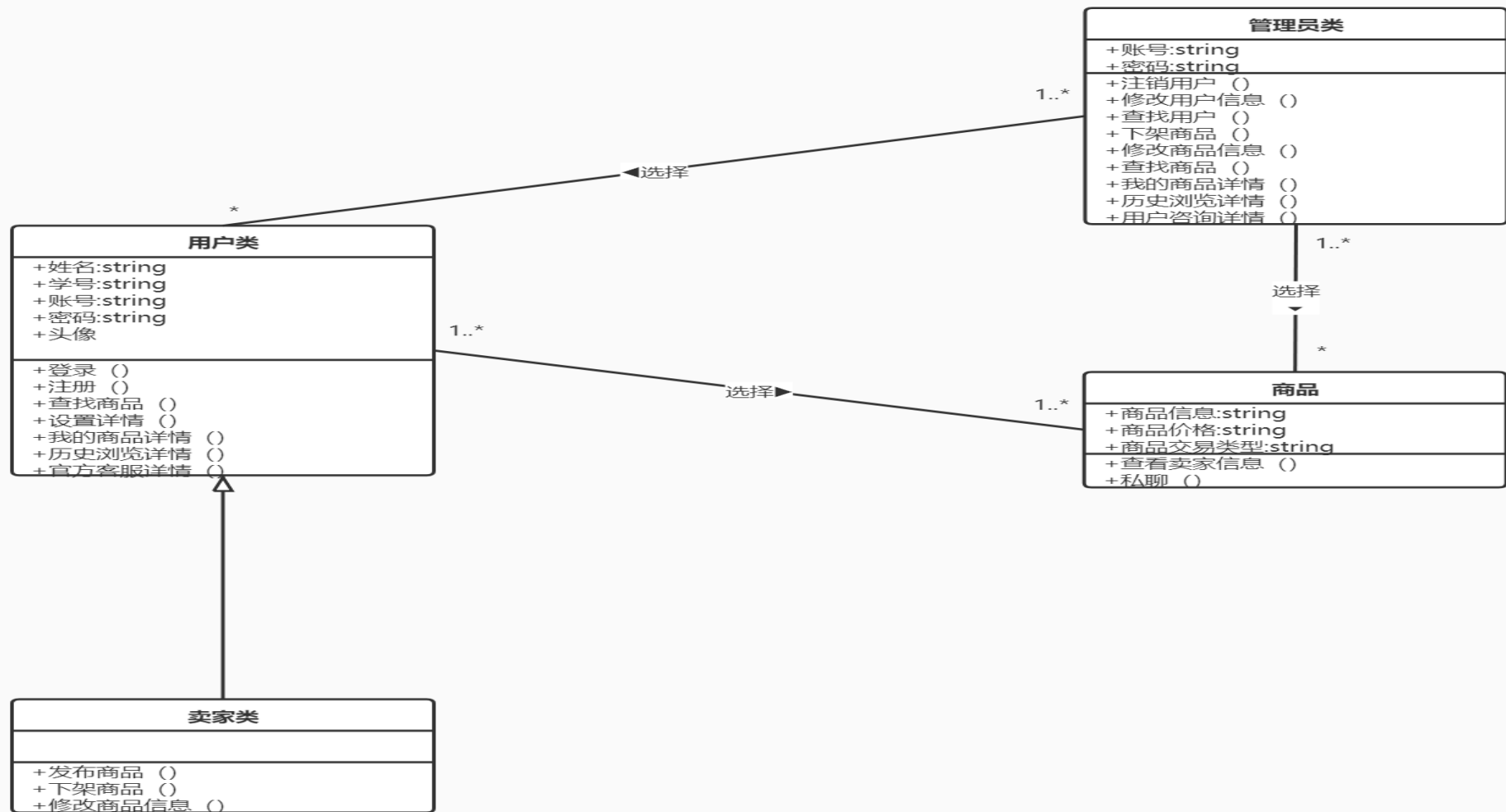
系统总体结构

- 5.安全设计:信息内部传输通过MD5不可逆加密算法。
- 6.小程序架构方面：微信小程序的架构包含两部分View视图层、App Service逻辑层。View层用来渲染页面结构。AppService层用来逻辑处理、数据请求、接口调用，它们在两个线程里运行。视图层和逻辑层通过系统层的JSBridge进行通信，逻辑层将数据进行处理后发送给视图层，同时接受视图层的事件反馈。视图层将逻辑层的数据反应成视图，同时将视图层的时间发送给逻辑层。在用户使用时，只允许用户输入我们期望的数据。
- 7.小程序后台管理方面：后端编写主要是用java，主要框架springboot，开发工具eclipse。要对原始数据进行加工和提取，为业务逻辑层提供数据服务。••••

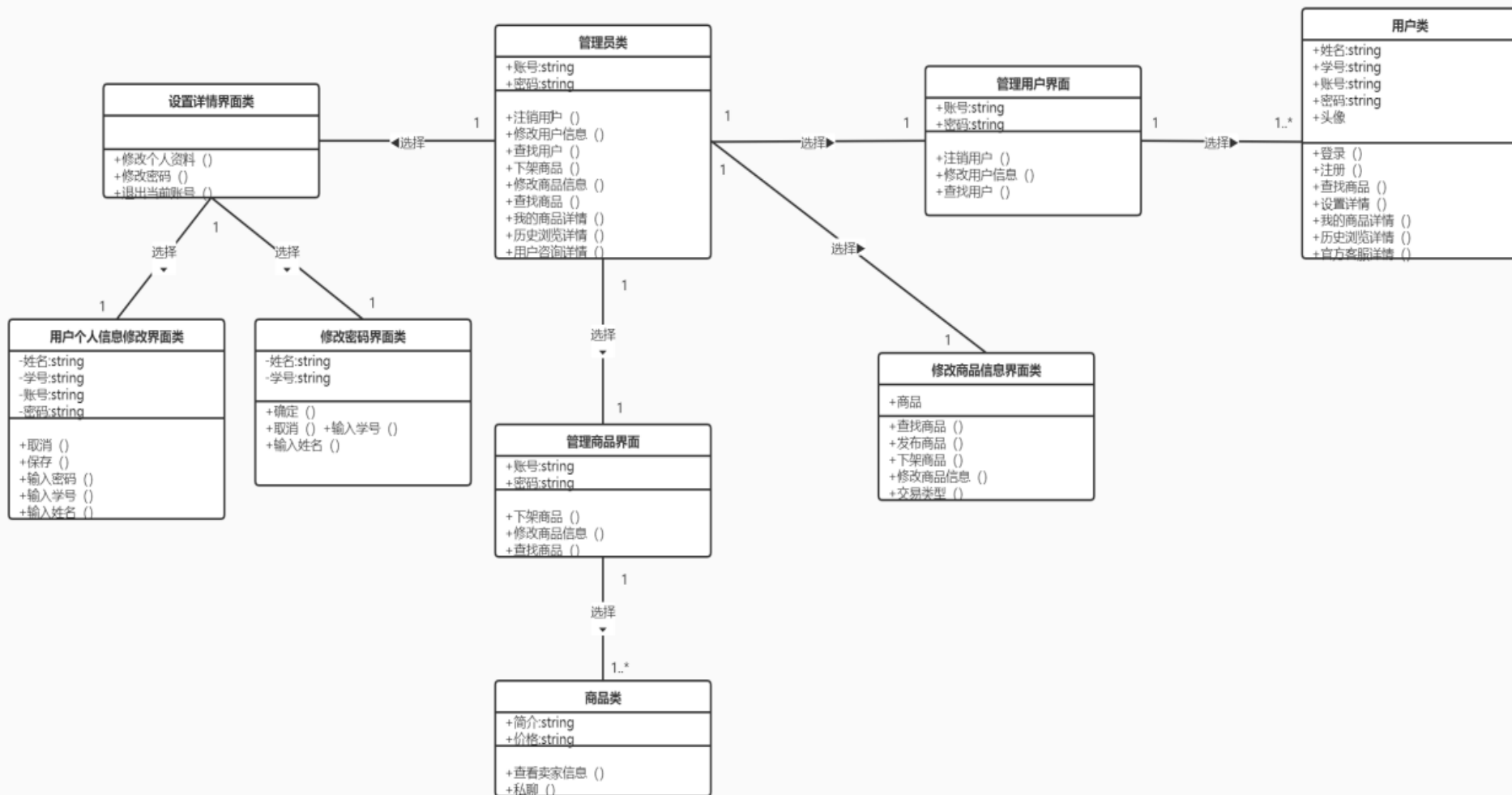
系统设计（类图）——用户界面



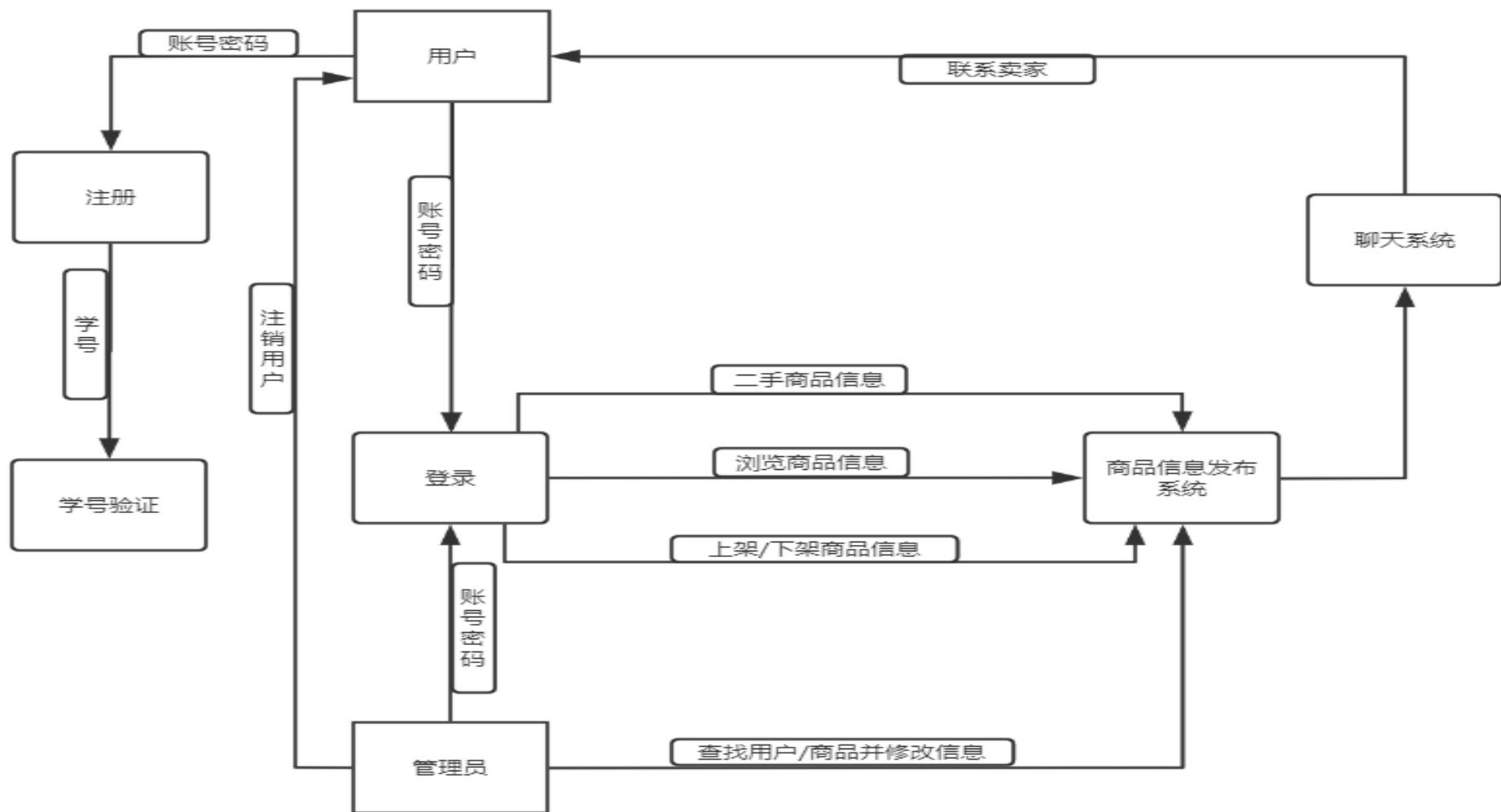
系统设计—实体类



系统设计—管理员界面



系统设计—数据流图



数据库设计

1. 具体设计

表名↵	功能说明↵
STUDENT↵	学生表，用于验证是否是学生↵
USER_INFO↵	用户信息表↵
COMMODITY↵	商品表，存放商品信息↵

2. STUDENT表 (学生表)

表名↵	STUDENT↵			
列名↵	数据类型(精度范围)↵	空/非空↵	约束条件↵	其他说明↵
STUDENT_ID↵	VARCHAR(22)↵	非空↵	PRIMARY KEY↵	学号↵
STUDENT_NAME↵	VARCHAR(40)↵	非空↵	↵	学生姓名↵
IS_REGISTER↵	VARCHAR(22)↵	↵	↵	是否已注册 (没注册为 空，已注册为 用户账号)↵

数据库设计

3. USER_INFO表 (用户信息表)

表名↵	USER_INFO↵			
列名↵	数据类型 (精度范围) ↵	空/非空↵	约束条件↵	其他说明↵
USER_ID↵	VARCHAR(22)↵	非空↵	PRIMARY KEY↵	账号↵
USER_NAME↵	VARCHAR(40)↵	非空↵	UNIQUE↵	用户名↵
STUDENT_ID↵	VARCHAR(40)↵	非空↵	↵	学号↵
USER_PASSWORD↵	VARCHAR(40)↵	非空↵	↵	密码↵
USER_PICTURE↵	IMAGE↵	↵	↵	头像(空的话, 用默认头像) ↵

4. COMMODITY表 (商品表)

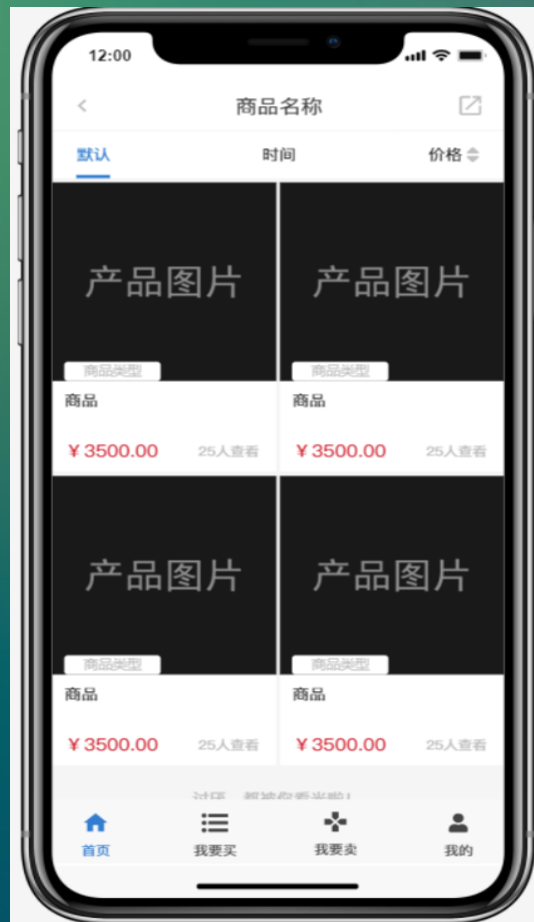
表名↵	COMMODITY↵			
列名↵	数据类型 (精度范围)↵	空/非空↵	约束条件↵	其他说明↵
COMMODITY_ID↵	VARCHAR(20)↵	非空↵	PRIMARY KEY↵	商品号↵
USER_ID↵	VARCHAR(22)↵	非空↵	↵	用户账号↵
COMMODITY_NAME↵	VARCHAR(40)↵	非空↵	↵	商品名↵
COMMODITY_INFO↵	VARCHAR(400)↵	非空↵	↵	商品信息↵
COMMODITY_PRICE↵	INT↵	非空↵	↵	商品价格↵
COMMODITY_PICTURE↵	IMAGE↵	非空↵	↵	商品图片↵

界面设计

未登录时:



登录后:



界面设计

登录后:

12:00

<

发布

* 商品名称:

请输入商品标题

* 价格:

请输入商品价格

* 交易类型

日常用品 >

效果图:

+

* 商品库存:

请输入商品库存

* 商品描述:

请输入商品描述

确认并发布


首页

我要买

我要卖


我的

12:00




子墨ZIMO


已实名




我的商品



历史浏览



官方客服



设置

首页

我要买

我要卖

我的

数据库设计部分

- 外部设计
- 结构设计
- 数据库设计
- 物理结构设计
- 运用设计

外部设计

1.标识符和状态

数据库软件的名称: MySql 5.0

数据库的名称为: CAMPUS_TRANSACTION_SQL

2.命名约定

所有的数据库命名都是以模块的英文名组成, 英文单词之间以下划线分开, 这样能够统一数据库表的命名, 也能够更好的规范数据库表命名。



外部设计

3.设计约定

在本系统中，数据库的设计采用PowerDesigner进行，并且采用面向对象的设计方法，首先进行对象实体的设计，最后将对象持久化到数据库中,所有的表和表之间的关联(ER图)都采用标准的PowerDesigner设计工具进行，这样能够将整个系统的设计和数据库设计有机的结合起来。



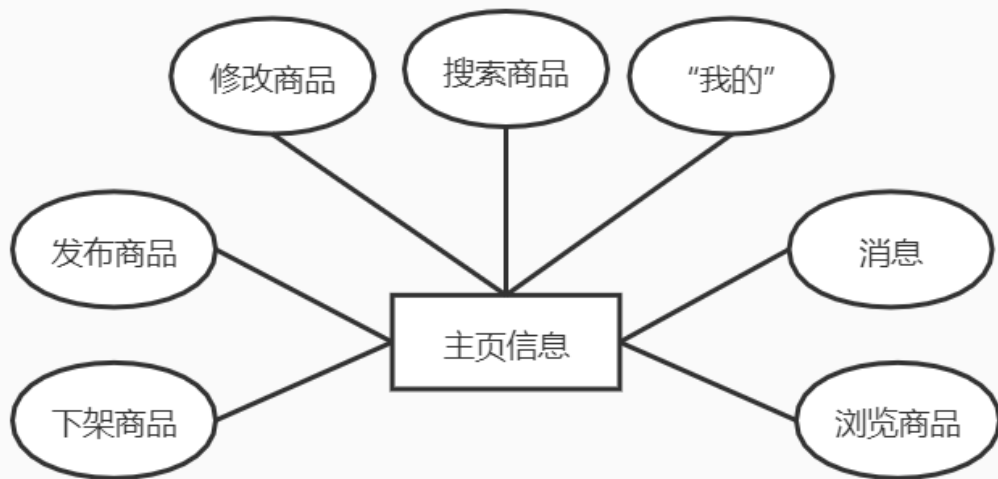
结构设计—概念结构设计

一.实体和属性的定义

1 用户模块：

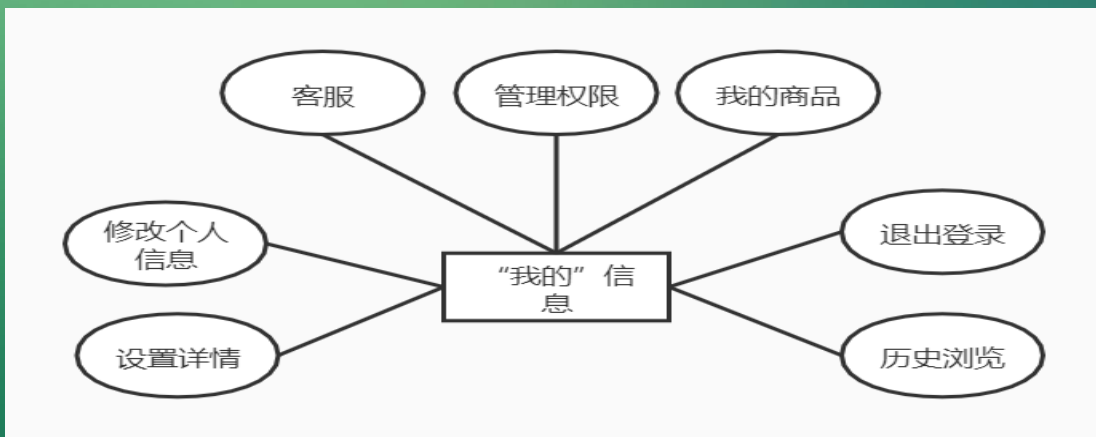


2. 程序详情模块：

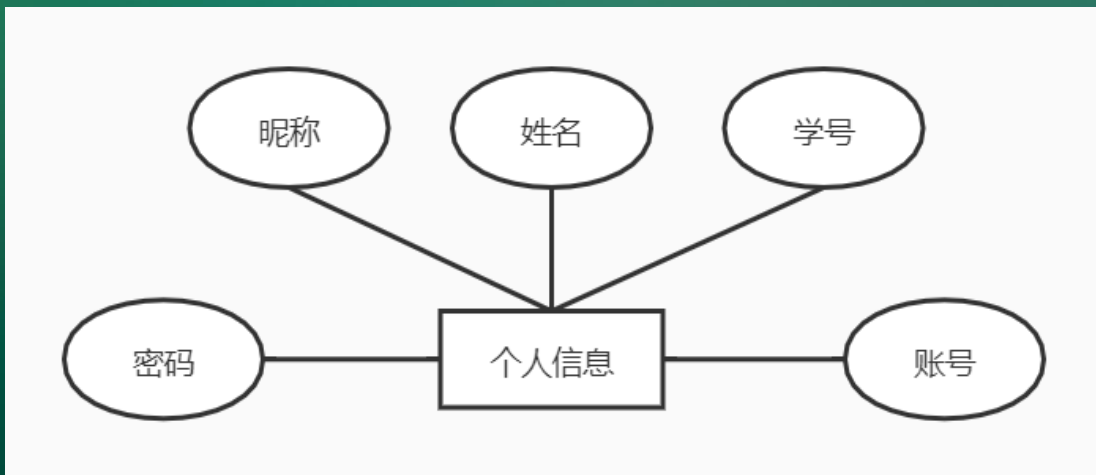


结构设计—概念结构设计

2. 程序详情模块:

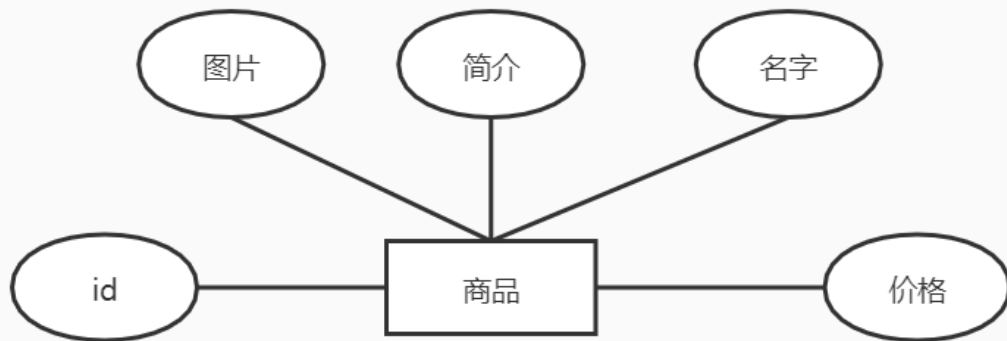


3. 信息模块:

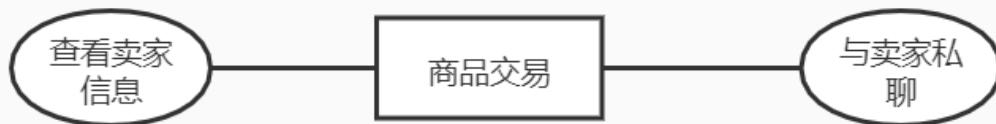


结构设计—概念结构设计

3. 信息模块：

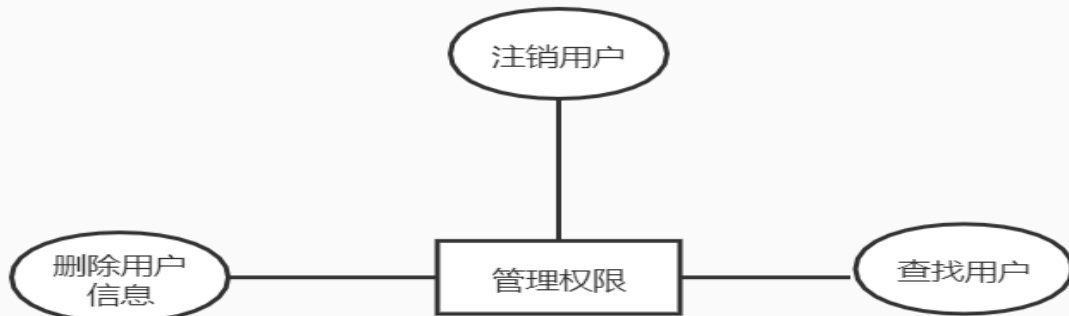


4. 交易模块：

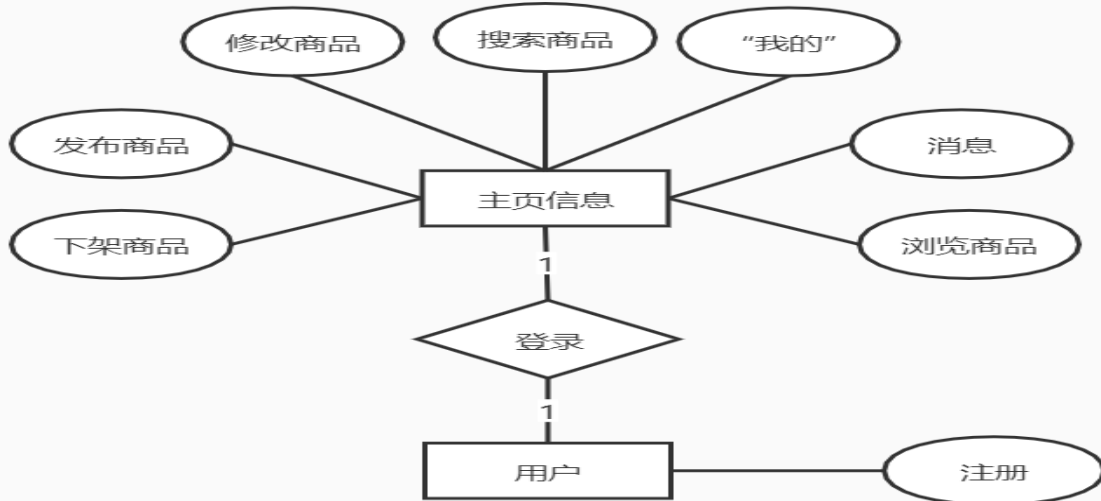


结构设计—概念结构设计

5. 管理模块

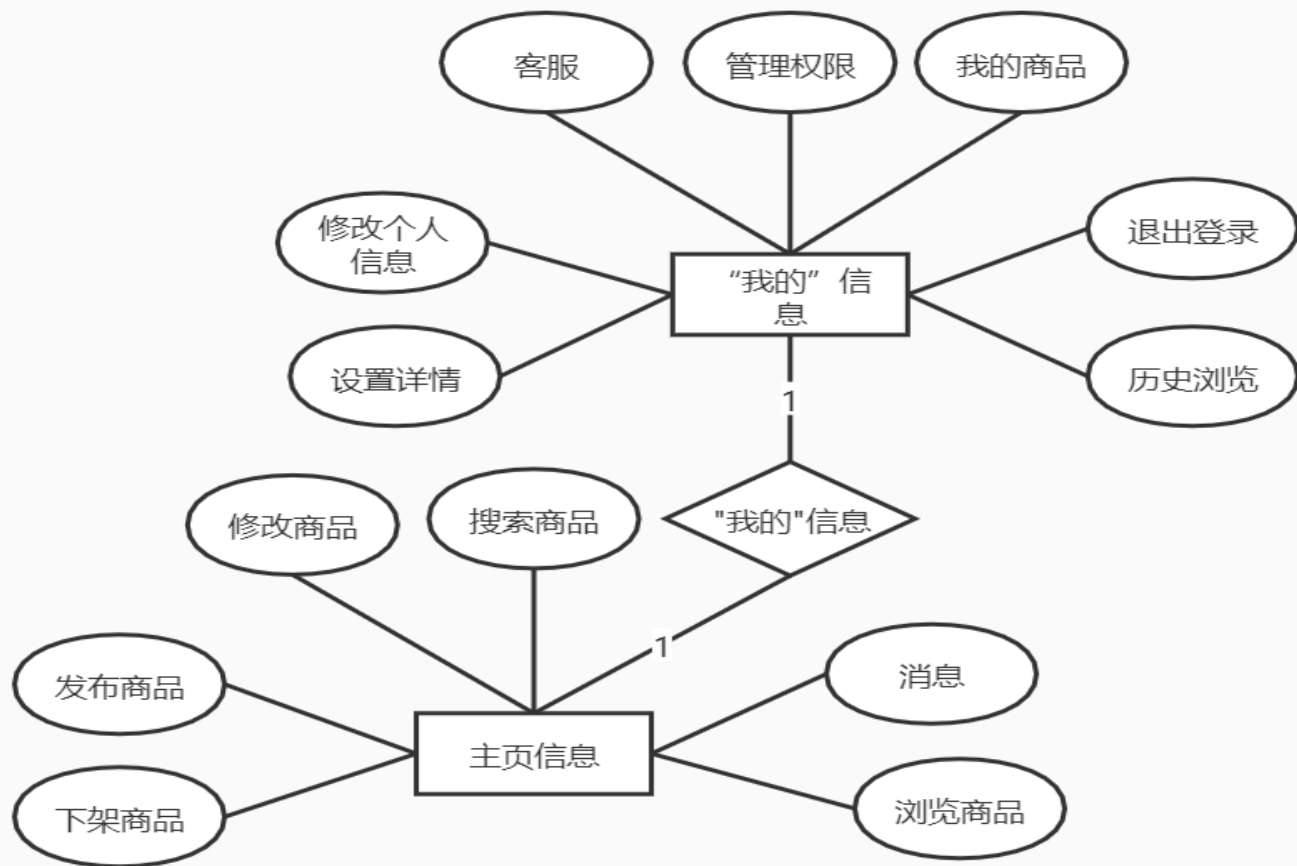


6. 登录局部E-R:



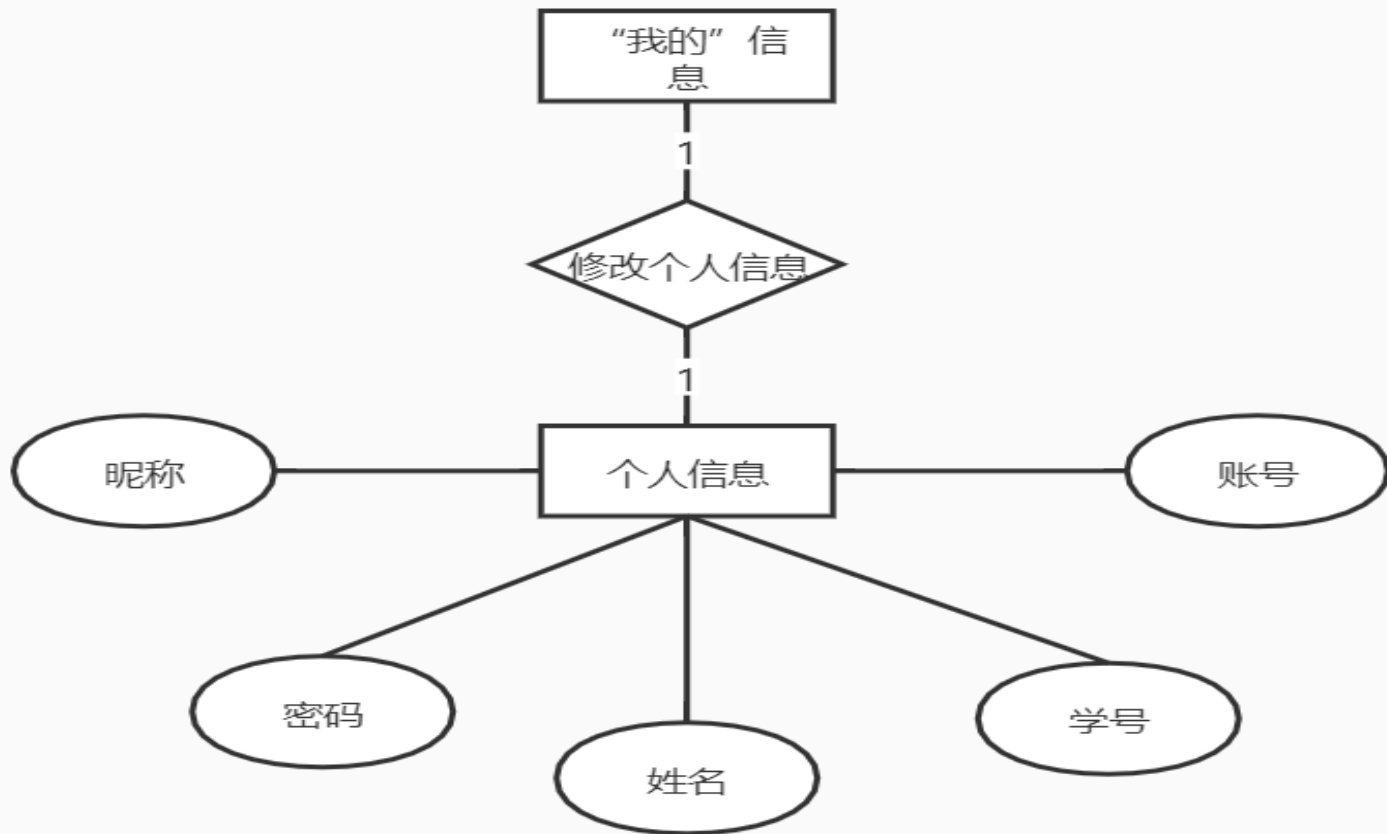
结构设计—概念结构设计

7.界面信息局部E-R:



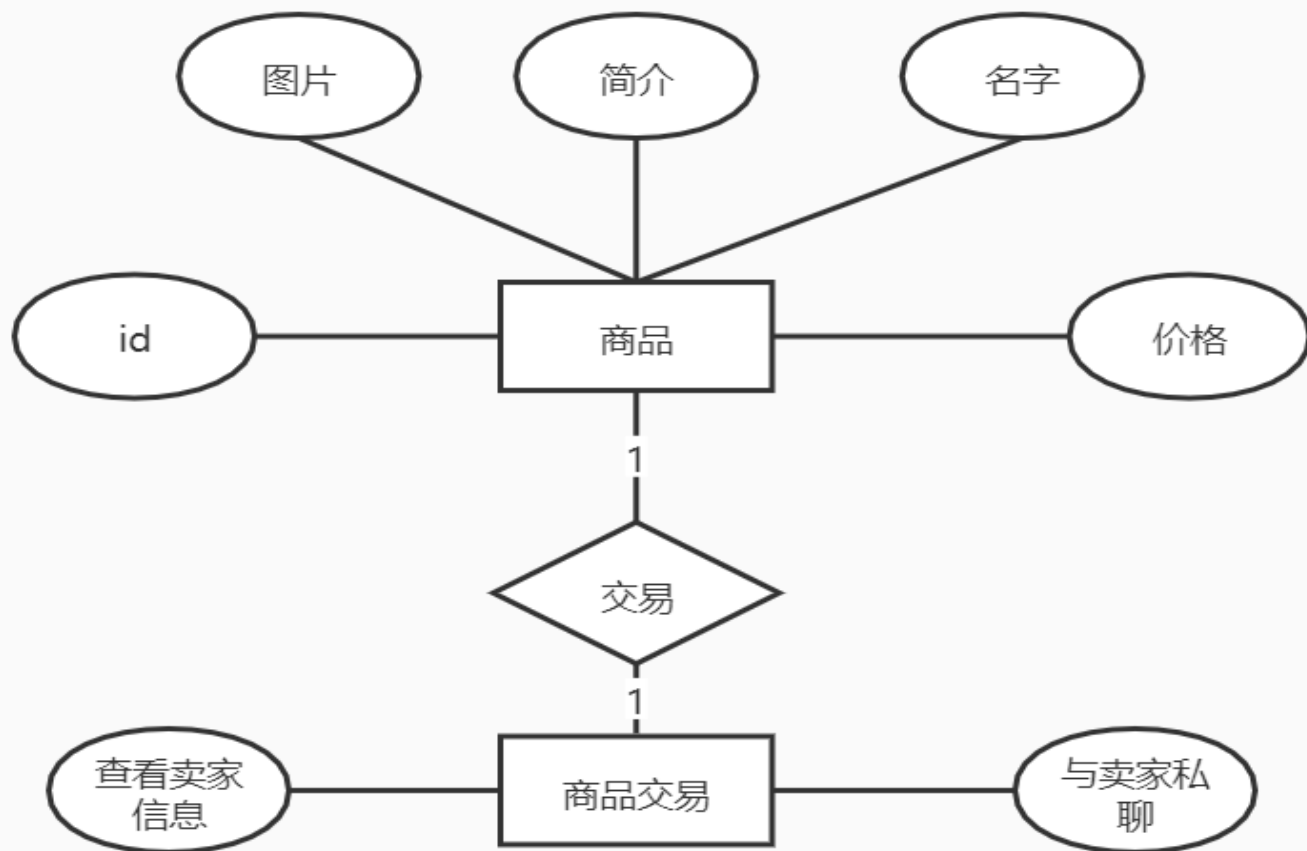
结构设计—概念结构设计

8.修改个人信息局部E-R:



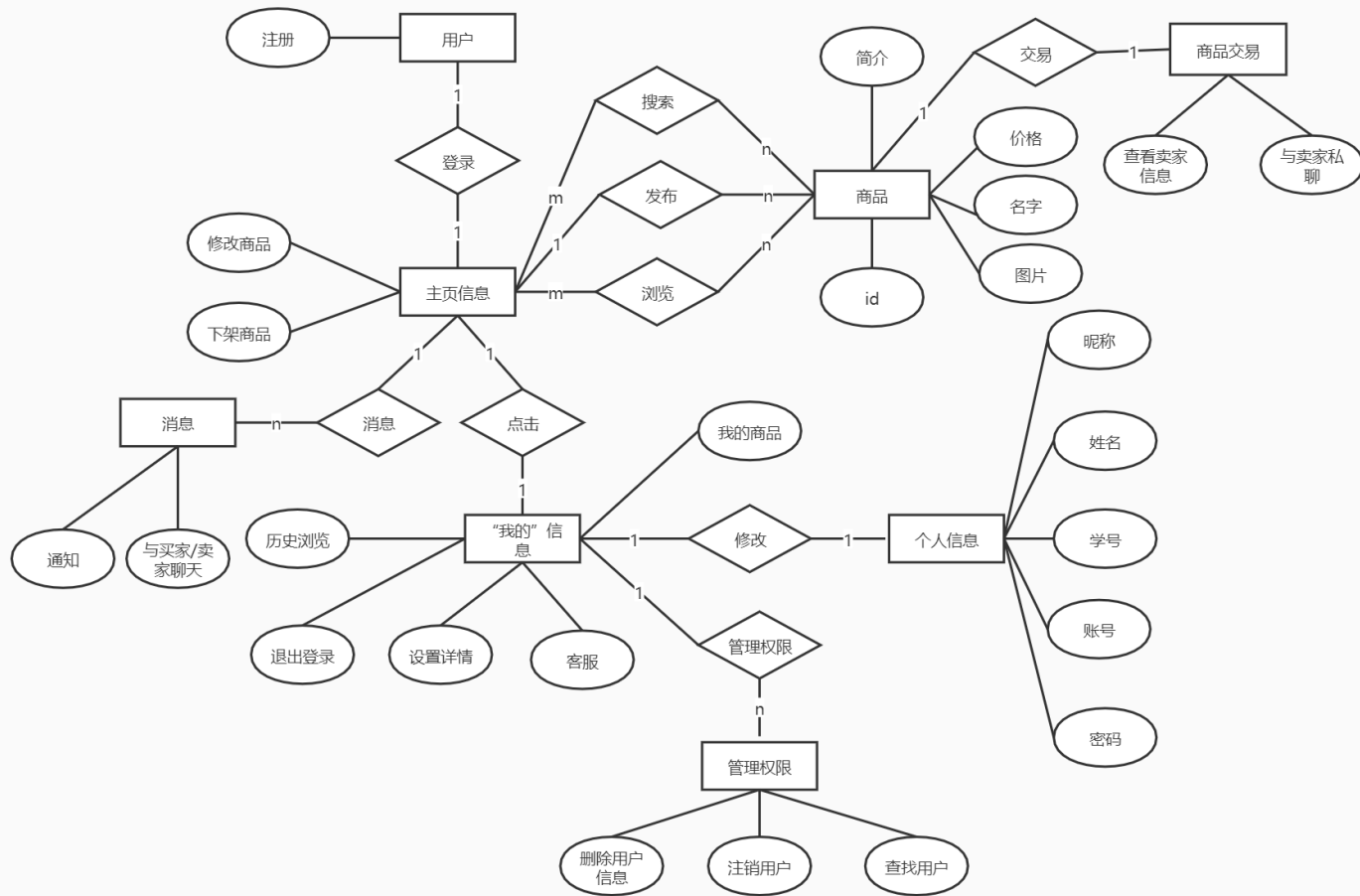
结构设计—概念结构设计

9.商品局部E-R:



结构设计—概念结构设计

9.全局E-R:



数据库设计

1. 具体设计

表名↵	功能说明↵
STUDENT↵	学生表，用于验证是否是学生↵
USER_INFO↵	用户信息表↵
COMMODITY↵	商品表，存放商品信息↵

2. STUDENT表 (学生表)

表名↵	STUDENT↵			
列名↵	数据类型(精度范围)↵	空/非空↵	约束条件↵	其他说明↵
STUDENT_ID↵	VARCHAR(22)↵	非空↵	PRIMARY KEY↵	学号↵
STUDENT_NAME↵	VARCHAR(40)↵	非空↵	↵	学生姓名↵
IS_REGISTER↵	VARCHAR(22)↵	↵	↵	是否已注册 (没注册为 空，已注册为 用户账号)↵

数据库设计

3. USER_INFO表 (用户信息表)

表名↵	USER_INFO↵			
列名↵	数据类型 (精度范围) ↵	空/非空↵	约束条件↵	其他说明↵
USER_ID↵	VARCHAR(22)↵	非空↵	PRIMARY KEY↵	账号↵
USER_NAME↵	VARCHAR(40)↵	非空↵	UNIQUE↵	用户名↵
STUDENT_ID↵	VARCHAR(40)↵	非空↵	↵	学号↵
USER_PASSWORD↵	VARCHAR(40)↵	非空↵	↵	密码↵
USER_PICTURE↵	IMAGE↵	↵	↵	头像(空的话, 用默认头像)↵

4. COMMODITY表 (商品表)

表名↵	COMMODITY↵			
列名↵	数据类型 (精度范围)↵	空/非空↵	约束条件↵	其他说明↵
COMMODITY_ID↵	VARCHAR(20)↵	非空↵	PRIMARY KEY↵	商品号↵
USER_ID↵	VARCHAR(22)↵	非空↵	↵	用户账号↵
COMMODITY_NAME↵	VARCHAR(40)↵	非空↵	↵	商品名↵
COMMODITY_INFO↵	VARCHAR(400)↵	非空↵	↵	商品信息↵
COMMODITY_PRICE↵	INT↵	非空↵	↵	商品价格↵
COMMODITY_PICTURE↵	IMAGE↵	非空↵	↵	商品图片↵

物理结构设计

数据库名称: CAMPUS_TRANSACTION_SQL

存储位置: 默认位置

建立系统程序员视图, 包括:

- a. 数据在内存中的安排, 包括对索引区、缓冲区的设计;
- b. 所使用的外存设备及外存空间的组织, 包括索引区、数据块的组织与划分;
- c. 访问数据的方式方法



运用设计—安全保密设计

通过区分不同的访问者、不同的访问类型和不同的数据对象，进行分别对待而获得的数据库安全保密设计考虑。对数据库设计中涉及到的各种项目，如数据项、记录、系、文卷、模式、子模式等一般要建立起数据保护措施，以说明它的标识符、同义名及有关信息。

数据库由专门数据库管理用员对数据库操作，需要注意以下几项安全问题：
访问安全、网络安全、传输安全、备份安全、数据安全



运用设计—数据库设计

创建表:

```
/*=====*/
/* DBMS name:      MySQL 5.0 */
/* Created on:     2020/4/19 23:26:26 */
/*=====*/

drop table if exists COMMODITY;

drop table if exists STUDENT;

drop table if exists USER_INFO;

/*=====*/
/* Table: COMMODITY */
/*=====*/

create table COMMODITY
(
    COMMODITY_ID      VARCHAR(20) not null,
    USER_ID           VARCHAR(22) not null,
    COMMODITY_NAME     VARCHAR(40) not null,
    COMMODITY_INFO     VARCHAR(400) not null,
    COMMODITY_PRICE    INT not null,
    COMMODITY_PICTRUE  IMAGE not null,
    primary key (COMMODITY_ID)
);
```

```
/*=====*/
/* Table: STUDENT */
/*=====*/

create table STUDENT
(
    STUDENT_ID        VARCHAR(22) not null,
    STUDENT_NAME       VARCHAR(40) not null,
    IS_REGISTER        VARCHAR(22),
    primary key (STUDENT_ID)
);

/*=====*/
/* Table: USER_INFO */
/*=====*/

create table USER_INFO
(
    USER_ID            VARCHAR(22) not null,
    USER_NAME           VARCHAR(40) not null,
    USER_PASSWORD       VARCHAR(40) not null,
    USER_PICTRUE        IMAGE,
    primary key (USER_ID),
    unique key UNQ_USER_INFO_USER_NAME (USER_NAME)
);

alter table COMMODITY add constraint FK_USERINFO_COMMODITY foreign key (USER_ID)
references USER_INFO (USER_ID) on delete restrict on update restrict;
```

关于老师提出的问题的回答

学生的学号信息如何更新？

学号信息因为平时有变动的概率不大，所以采取每年新生入校时更新数据库，确保新生能够注册。



计划开发时间安排

预计开发时间表：

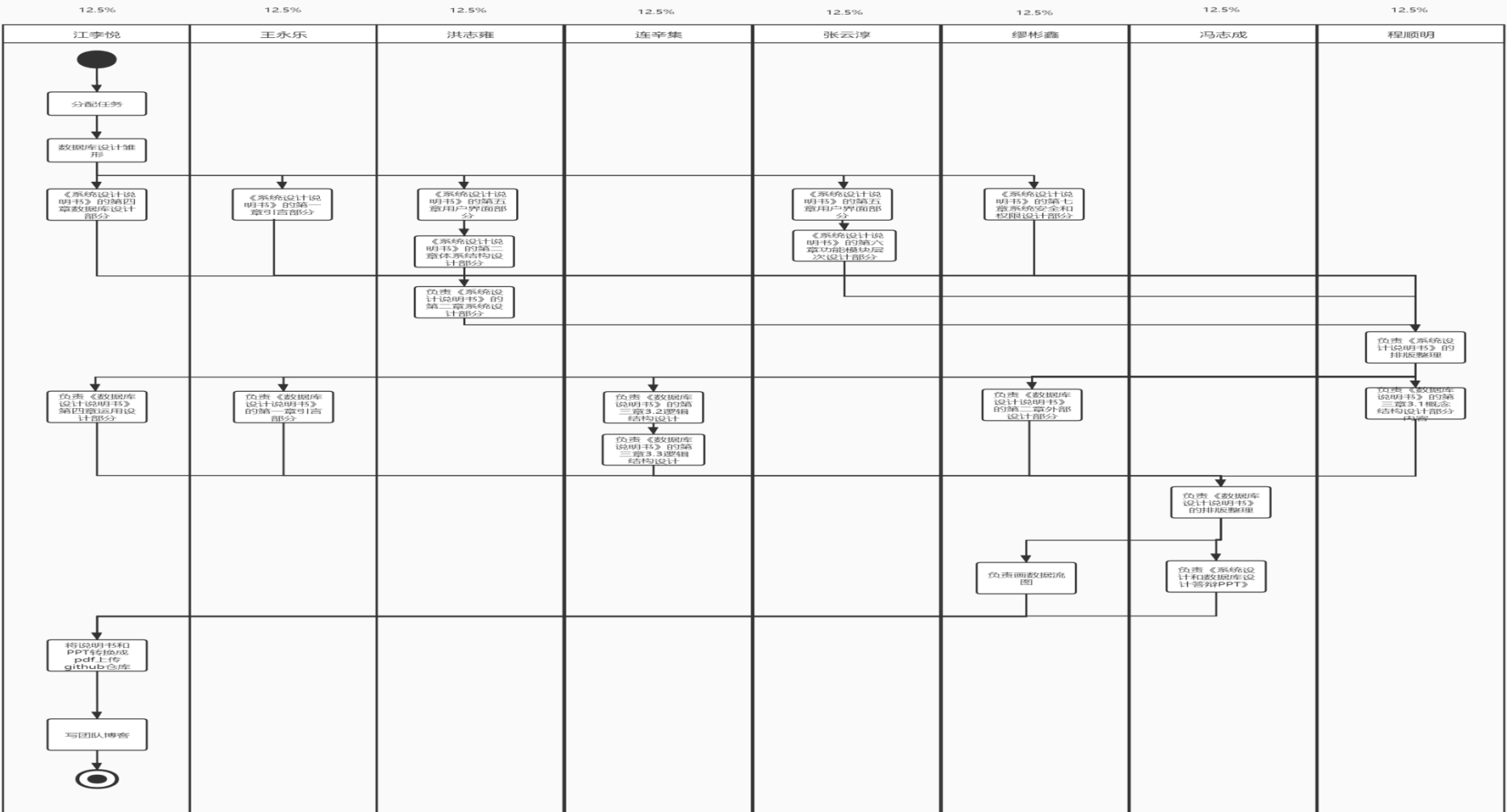
起始时间	任务安排	截止时间
4月13日	1.制作系统设计说明书 2.制作数据库设计说明书 3.制作系统设计和数据库设计答辩PPT	4月19日
4月20日	前端后端同时开始开发	4月26日
4月27日	前端后端完成简易版本的《校易》	5月3日
5月4日	调试软件，修复发现的bug	5月10日
5月11日	优化《校易》界面和代码	5月17日
5月18日	调试软件，修复新的bug	5月24日
5月25日	优化《校易》界面和代码	5月31日
6月1日	调试软件，修复新的bug	6月7日

计划开发分工安排

预期分工：

组员	角色	负责的开发部分
洪志雍	前端	1.调用接口，和后端对接 2.协助设计一些css,js样式
张云淳	前端	1.界面设计和排版 2.负责大部分css和js
江李悦	后端	1.搭建服务器 2.创建数据库 3.完成功能模块中的管理员模块
程顺明	后端	1.完成功能模块中的用户注册登录模块 2.完成功能模块中的用户个人设置模块
冯志成	后端	完成功能模块中的用户商品上架下架模块
连辛集	后端	完成功能模块中的用户查询模块
缪彬鑫	后端	完成功能模块中的用户聊天交流模块
王永乐	测试	测试《校易》，寻找其中的bug和不足，以及对软件提出改进意见

此次答辩小组分工泳道图



—— THANK YOU! ——

码到成功小组制作

