

数据库课程设计报告

-----超市会员系统

软件学院

软件工程专业

2023级6班

姓名：张岩

学号：202300300046

任课教师：闫中敏

实验教师：闫中敏

目 录

一、 系统开发平台	4
二、数据库规划	4
2.1 任务陈述.....	4
2.2 任务目标	4
三、系统定义	5
3.1 系统边界	5
3.2 用户视图	6
四、需求分析	6
4.1 用户需求说明	6
4.1.1 数据需求.....	6
4.1.2 事务需求.....	7
4.2 系统需求说明	8
五、数据库逻辑设计	9
5.1 E-R 图	9
5.2 数据字典	9
5.3 关系表	12
六、数据库物理设计	17
6.1 索引	17

6.2 安全机制	17
七、应用程序设计	18
7.1 功能模块	18
7.2 界面设计	19
7.3 事务设计	27
7.3.1 新用户注册.....	27
7.3.2 用户修改密码和充值.....	28
7.3.3 购买商品.....	30
7.3.4 兑换满减券.....	33
7.3.5 商品修改.....	35
7.3.6 充值金额统计.....	37
7.3.7 消费金额统计.....	37
八、测试和运行	38
8.1 登录	38
8.2 管理员	39
8.3 用户	41
九、总结	44

一、系统开发平台

题目：超市会员系统

开发工具：IntelliJ IDEA , Visual Studio Code

技术栈：Spring Boot 和 Vue3.0

数据库：MySQL

操作系统：Windows 11

二、数据库规划

2.1 任务陈述

火车票售票系统是为了让用户能够在线上进行车票的预定，从而避免线下排队和信息不公开的弊端。

本系统是一个综合性的票务管理系统，主要功能包括火车票、飞机票、汽车票的查询、预订、退票、改签等操作。系统采用前后端分离的架构设计，为用户提供便捷的在线购票服务，同时为管理员提供完善的后台管理功能。

系统开发的总体任务是实现火车票各种信息的系统化、规范化和自动化。

2.2 任务目标

用户功能目标：

- 用户注册、登录、个人信息管理
- 满减券、商品的查询与兑换（购买）
- 订单管理（查看购买物品）

-流水查询（查看积分与充值金额的动态）

管理员功能目标

-会员管理（查看会员动态、修改会员密码）

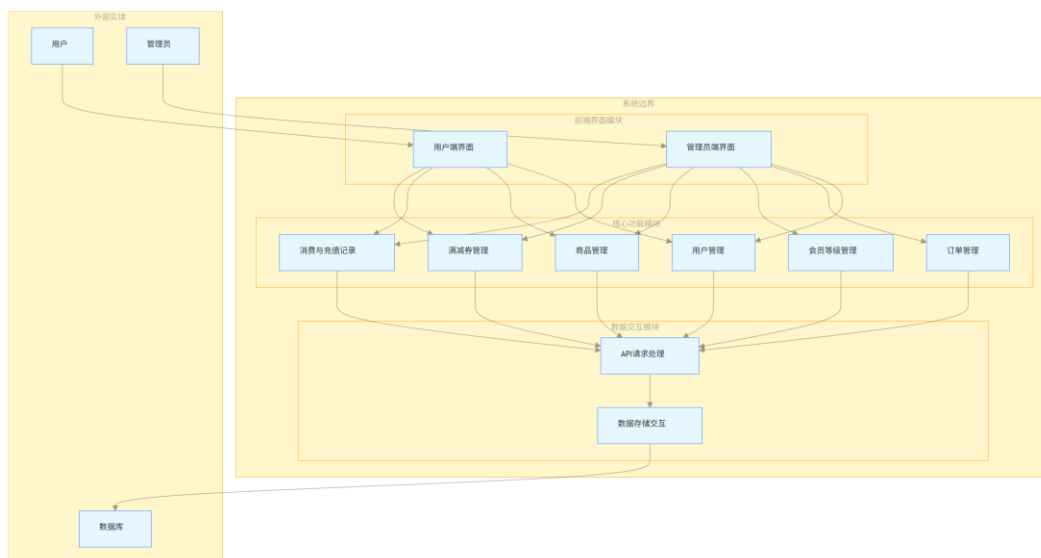
-订单管理

-商品管理

-系统数据统计

三、系统定义

3.1 系统边界



3.2 用户视图

用户	需求
User(用户)	注册（成为新用户），商品基本信息查询、购买，满减券信息查询、兑换，订单查询，流水查询，个人信息维护和密码修改。
Admin(管理员)	登录，商品基本信息的管理，订单的管理，用户信息的管理，当天营业额和总营业额的统计。

四、需求分析

4.1 用户需求说明

4.1.1 数据需求

用户数据：

- 用户基本信息：账号、密码、姓名、角色、电话、余额、积分、会员等级、身份证号
- 用户认证信息：登录状态、权限级别

商品和满减券数据：

- 商品信息：ID、商品名称、价格、评分、商品描述、商品种类（标签）
- 折扣券信息：种类、满减金额、最低使用金额、兑换所需积分

订单数据：

-订单基本信息：订单 ID、用户 ID、下单时间、花费金额、使用的满减券信息、订单详情

- 订单详情：ID、订单 ID、商品 ID、商品数量、原始金额、实际金额

其他数据：

- 兑换信息：ID、兑换数量、满减券种类、用户 ID、兑换时间、花费积分

- 会员等级信息：种类、名称、所需金额、折扣

- 充值信息：ID、用户 ID、充值数目、充值时间

4.1.2 事务需求

数据录入

1. 录入用户的基本信息
2. 录入管理员的基本信息
3. 录入商品和满减券的基本信息
4. 录入会员等级的基本信息

数据更新/删除

5. 更新/删除用户及用户信息
6. 更新/删除商品的信息
7. 更新/删除满减券的信息

数据查询

数据库必须支持以下查询：

User(用户)： 查询任意商品以及满减券的基本信息

查询任意日期的订单的信息

查询过去流水的信息

查询个人基本信息和密码

Admin(管理员): 查询任意商品基本信息

查询任意日期的订单的信息

查询过去流水的信息

查询营业的总销售额

4.2 系统需求说明

初始数据库大小

1. 大约共有 50 种商品和 50 中满减券
2. 大约有 50 个用户，一个管理员
3. 大约有 200 条购买记录以及 250 条兑换记录
4. 大约有 60 条充值记录
5. 大约有 50 个会员等级

性能需求

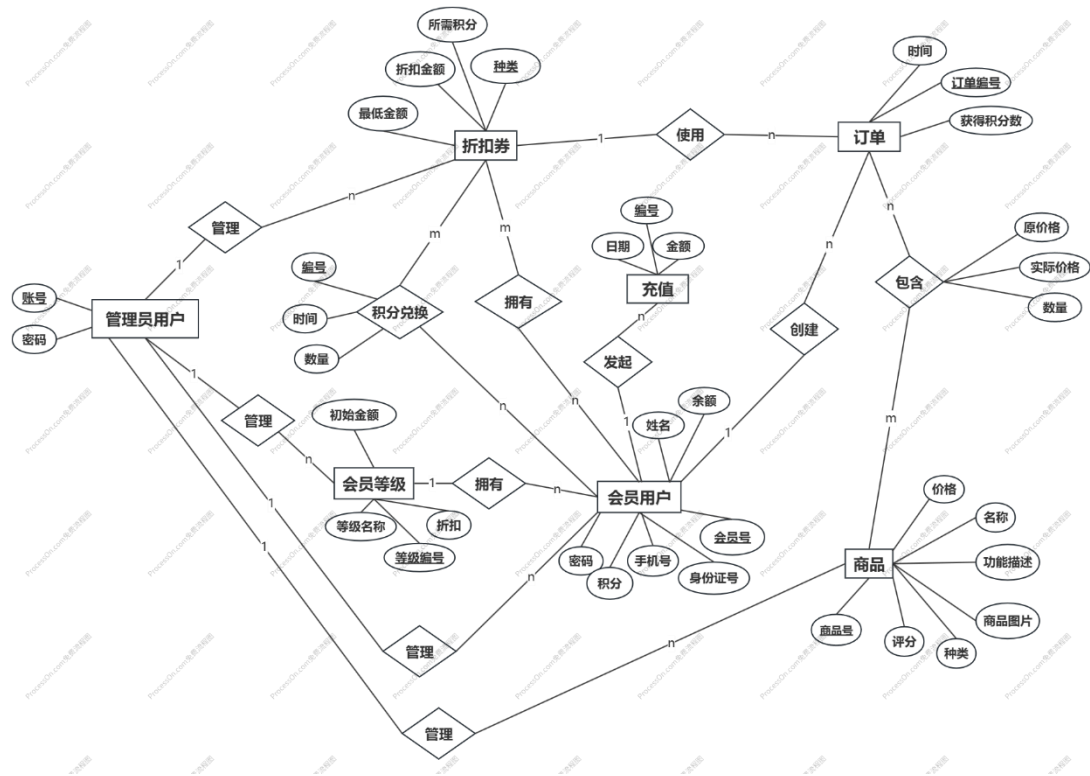
1. 响应时间：页面加载时间小于 2 秒
2. 并发用户：支持 1000+并发用户
3. 可用性：系统可用性大于 99%

安全需求

1. 用户认证：密码加密存储
2. 权限控制：基于角色的访问控制
3. 数据安全：敏感数据加密传输

五、数据库逻辑设计

5.1 E-R 图



5.2 数据字典

实体	属性	注释	类型	长度	键	能否空	约束
管理员	account	管理员账号	Varchar	16	主键	N	
	password	密码	varchar	64		N	
用户	account	会员账号	int unsigned	8	主键	N	数字表示

	name	会员姓名	varchar	30		N	
	password	密码	varchar	15		N	
	points	积分	int			N	
	phone_number	电话	char	11		N	
	ID_number	身份证号	char	18		N	
	Level	VIP 等级	int unsigned	1	外键	N	
	balance	余额	double	0		N	不能为负

实体	属性	注释	类型	长度	键	能否空	约束
商品	id	ID	int unsigned		主键	N	数字表示
	name	商品名称	varchar	255		N	
	price	价格	double	0		N	
	rating	商品评分	double unsigned	2		N	

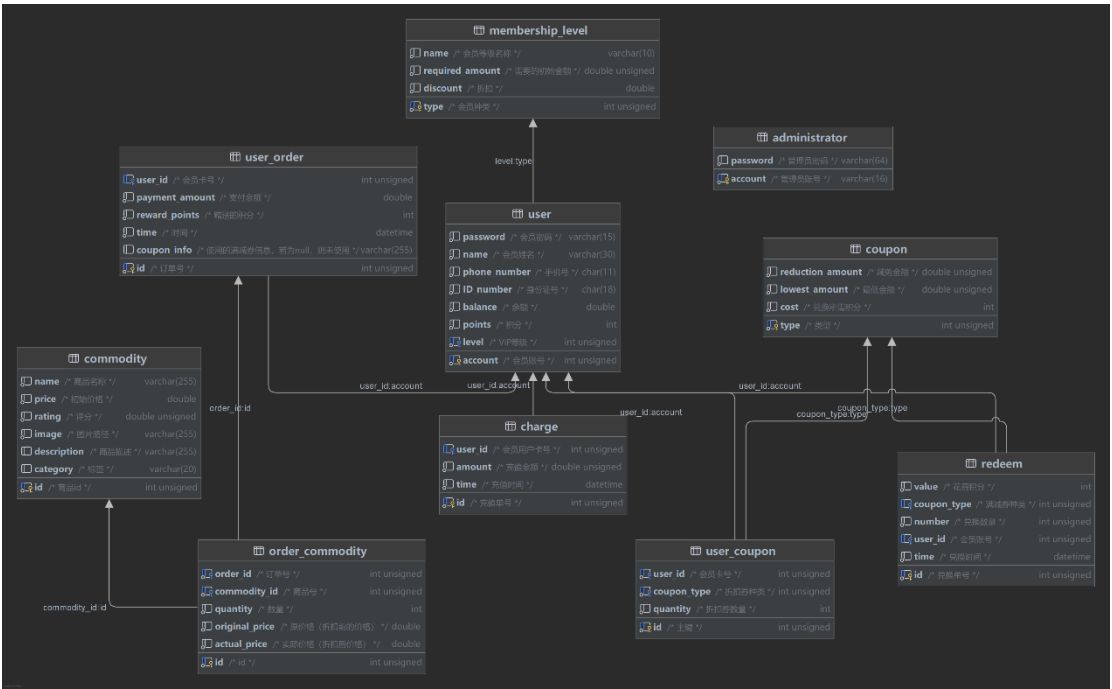
	image_path	商品图片路径	varchar	255		N	
	description	商品描述	varchar	255		Y	
	category	标签	Varchar	20		Y	

实体	属性	注释	类型	长度	键	能否空	约束
商品 订单	id	ID	int unsigned		主键	N	数字
	user_id	用户 ID	int	8	外键	N	
	time	下单时间	datetime			N	
	payment_amount	支付金额	double			N	
	reward_points	赠送的积分	int			N	
	coupon_info	使用的满减券信息	varchar	255		Y	
实体	属性	注释	类型	长度	键	能否空	约束

满减券	type	类型	int unsigned		主键	N	
	reduction_amount	减免金额	double unsigned			N	
	lowest_amount	最低金额	double unsigned			N	
	cost	兑换所需积分	int			N	

注：这里省略了兑换记录、充值记录，以及会员等级的表格。

5.3 关系表



创建管理员表

```
create table administrator

(

    account    varchar(16) not null comment '管理员账号'

        primary key,

    password varchar(64) not null comment '管理员密码'

);
```

创建用户表

```
DROP TABLE IF EXISTS `user`;

create table user

(

    account      int unsigned auto_increment comment '会员账号'

        primary key,

    password     varchar(15)                not null comment '会员密码',

    name         varchar(30)                not null comment '会员姓名',

    phone_number char(11)                  not null comment '手机号',

    ID_number    char(18)                  not null comment '身份证号',

    balance      double      default 0      not null comment '余额',

    points       int          default 0      not null comment '积分',

    level        int unsigned default '1' not null comment 'VIP 等级',

    constraint user_ibfk_1

        foreign key (level) references membership_level (type)

);
```

创建商品信息表

```
create table commodity

(

    id          int unsigned auto_increment comment '商品 id'
```

```
        primary key,

    name          varchar(255)          not null comment '商品名称',

    price         double                 not null comment '初始价格',

    rating        double unsigned default '0' not null comment '评分',

    image         varchar(255)           not null comment '图片路径',

    description   varchar(255)           null comment '商品描述',

    category      varchar(20)            null comment '标签'

);
```

创建满减券表

```
create table coupon

(

    type          int unsigned auto_increment comment '类型'

        primary key,

    reduction_amount double unsigned not null comment '减免金额',

    lowest_amount  double unsigned not null comment '最低金额',

    cost          int                not null comment '兑换所需积分'

);
```

创建会员等级表

```
create table membership_level

(

    type          int unsigned auto_increment comment '会员种类'

        primary key,

    name          varchar(10)          not null comment '会员等级名称',

    required_amount double unsigned not null comment '需要的初始金额',

    discount       double              not null comment '折扣'

);
```

创建充值记录表

```

create table charge

(
    id          int unsigned auto_increment comment '充值单号'
        primary key,
    user_id int unsigned          null comment '会员用户卡号',
    amount  double unsigned not null comment '充值金额',
    time    datetime             not null comment '充值时间',
    constraint charge_ibfk_1
        foreign key (user_id) references user (account)
        on update cascade
);

```

创建兑换记录表

```

create table redeem

(
    id          int unsigned auto_increment comment '兑换单号'
        primary key,
    value       int              not null comment '花费积分',
    coupon_type int unsigned null comment '满减券种类',
    number      int unsigned not null comment '兑换数量',
    user_id     int unsigned null comment '会员账号',
    time        datetime        not null comment '兑换时间',
    constraint redeem_ibfk_1
        foreign key (coupon_type) references coupon (type)
        on update cascade,
    constraint redeem_ibfk_2
        foreign key (user_id) references user (account)
        on update cascade
);

```

创建会员——满减券表

```
create table user_coupon
(
    id            int unsigned auto_increment comment '主键'
        primary key,
    user_id       int unsigned not null comment '会员卡号',
    coupon_type   int unsigned not null comment '折扣券种类',
    quantity      int default 0 not null comment '折扣券数量',
    constraint user_coupon_ibfk_1
        foreign key (coupon_type) references coupon (type)
            on update cascade,
    constraint user_coupon_ibfk_2
        foreign key (user_id) references user (account)
            on update cascade
);
```

创建会员——订单表

```
create table user_order
(
    id            int unsigned auto_increment comment '订单号'
        primary key,
    user_id       int unsigned null comment '会员卡号',
    payment_amount double not null comment '支付金额',
    reward_points int not null comment '赠送的积分',
    time          datetime not null comment '时间',
    coupon_info    varchar(255) null comment '使用的满减券信息，若为 null，则未使用',
    constraint user_order_ibfk_1
        foreign key (user_id) references user (account)
```



```
on update cascade  
  
);
```

创建订单详情表

```
create table order_commodity  
(  
    id int unsigned auto_increment comment 'id'  
        primary key,  
    order_id int unsigned not null comment '订单号',  
    commodity_id int unsigned not null comment '商品号',  
    quantity int not null comment '数量',  
    original_price double not null comment '原价格（折扣前的价格）',  
    actual_price double not null comment '实际价格（折扣后价格）',  
    constraint order_commodity_ibfk_1  
        foreign key (order_id) references user_order (id)  
            on update cascade,  
    constraint order_commodity_ibfk_2  
        foreign key (commodity_id) references commodity (id)  
            on update cascade  
);
```

六、数据库物理设计

6.1 索引

1. 主键索引

- 所有表的主键自动创建聚簇索引

6.2 安全机制

1. 用户权限控制：

基于角色的访问控制 (RBAC)

普通用户：只能访问自己的数据

管理员：可以访问所有数据

2. 数据安全

敏感数据传输使用 HTTPS

使用 **Tokenization**（令牌化）技术，保证数据安全

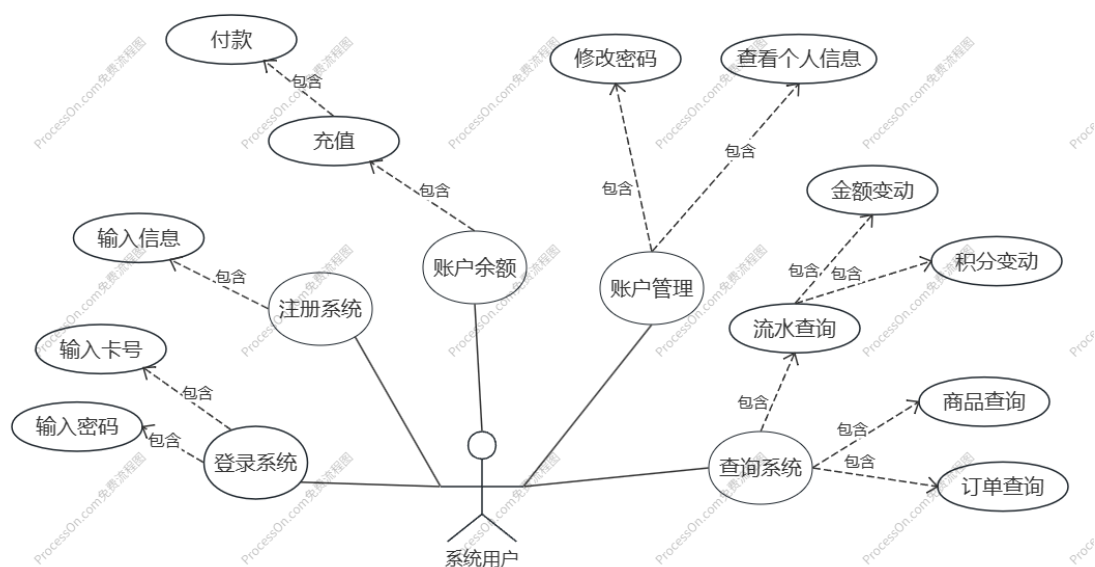
3. 操作审计

关键操作记录日志

用户登录记录

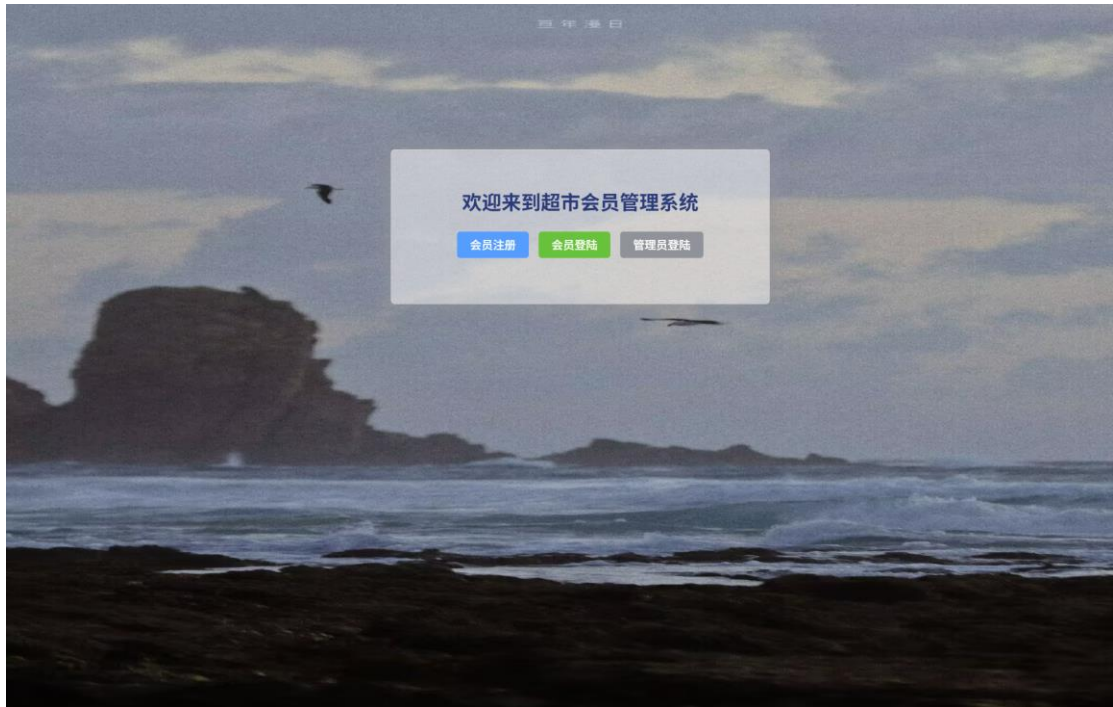
七、应用程序设计

7.1 功能模块

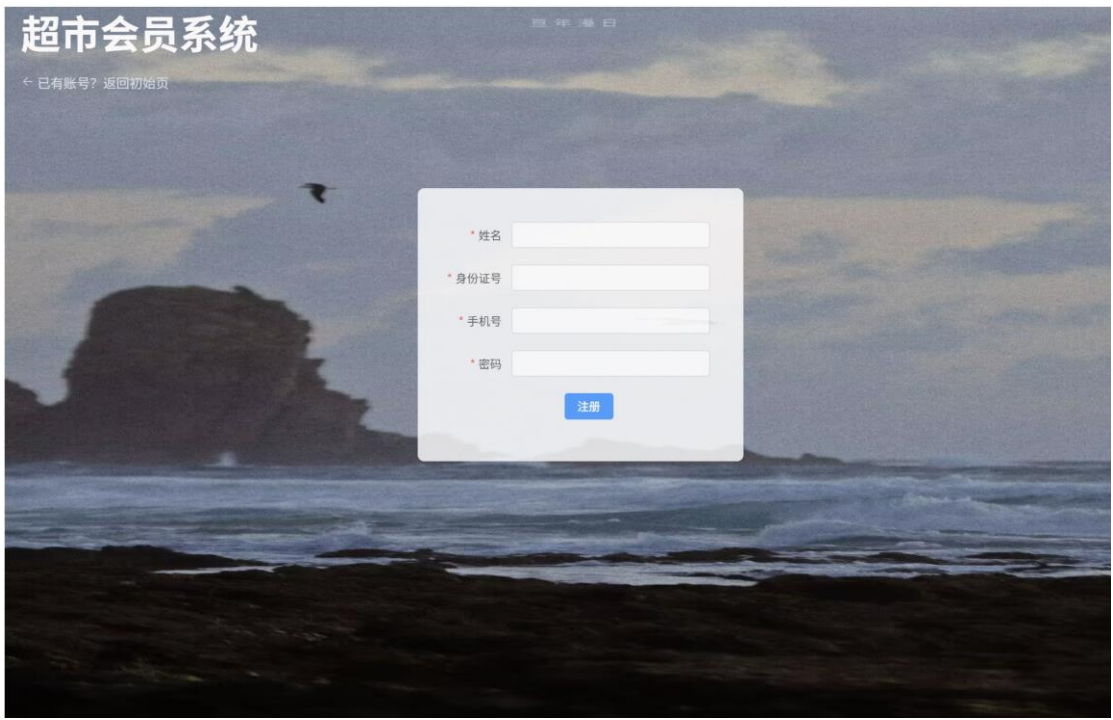


7.2 界面设计

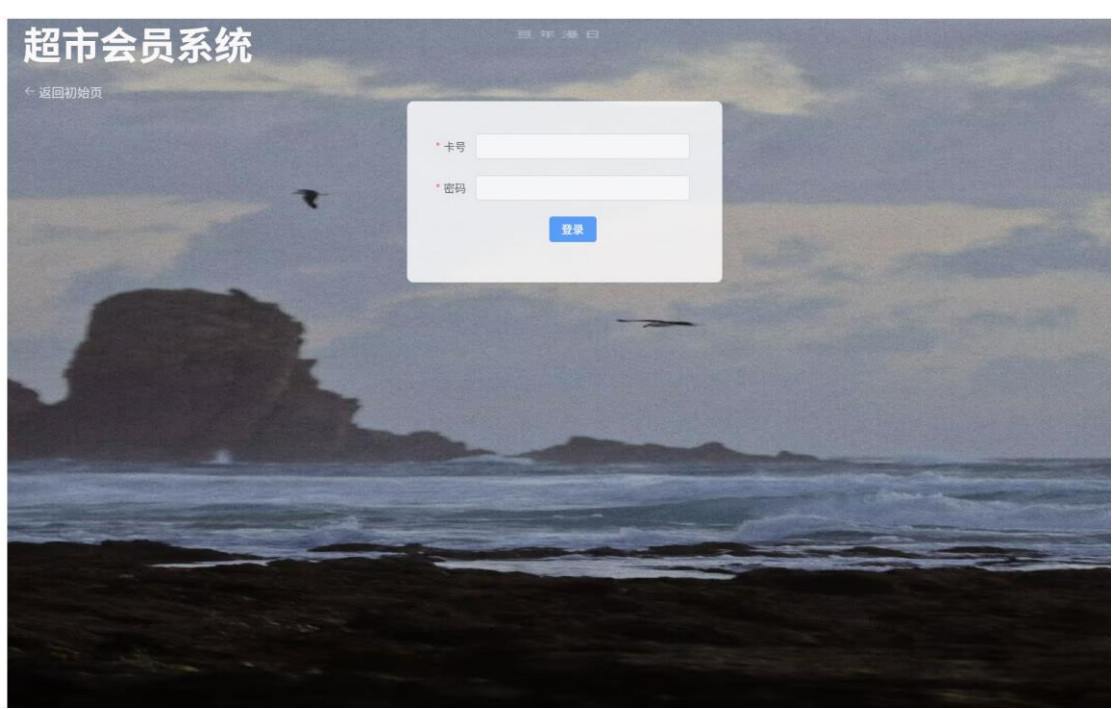
1. 登陆和注册界面



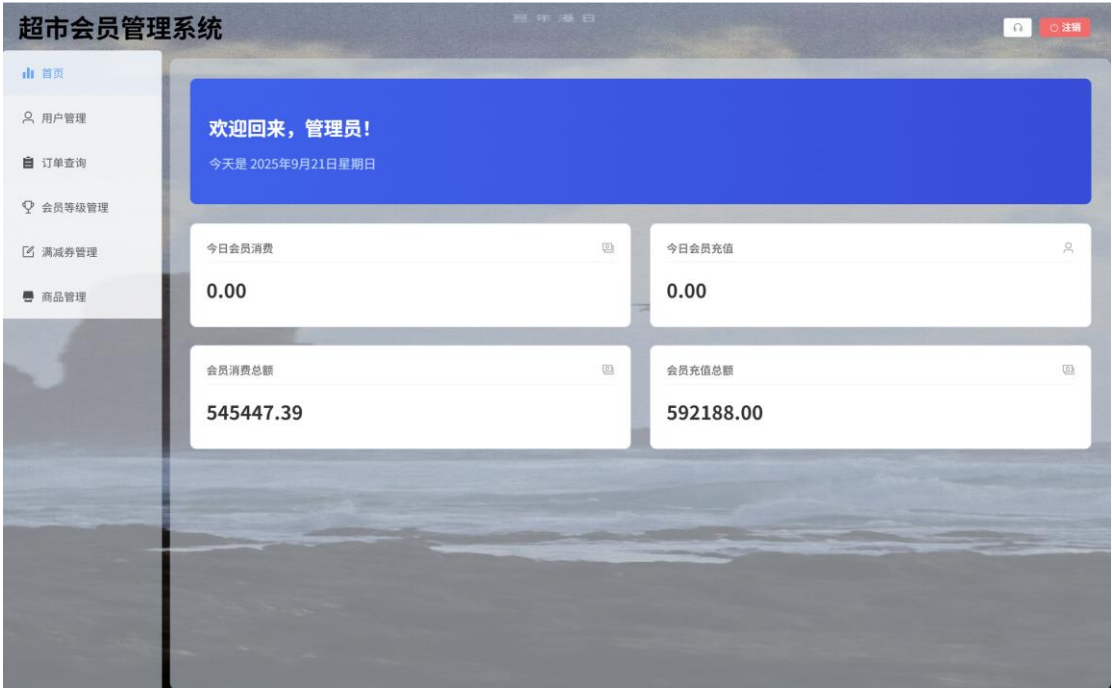
注册页面：



登录页面：



2. 管理员界面



超市会员管理系统

2025年9月21日

注册

首页

用户管理

订单查询

会员等级管理

满减券管理

商品管理

用户管理

卡号	姓名	手机号	身份证号	余额	积分	会员等级	操作
10000052	njbwe	13184225581	147*****3	888.00	0	1	查看动态 修改信息
10000051	mknqdw	13184225523	147*****4	113.00	10	2	查看动态 修改信息
10000050	mkwe	13184225584	147*****4		0	1	查看动态 修改信息
10000049	nkonerg	13800148000	147*****4		0	1	查看动态 修改信息
10000048	noaedi	13800148000	147*****4		0	1	查看动态 修改信息
10000047	nuorg	13800148000	123*****6		0	1	查看动态 修改信息
10000046	mkegreg	13800148000	123*****5	79.00	5651	1	查看动态 修改信息
10000045	bjwef	13800148000	147*****7	59.00	815	1	查看动态 修改信息

超市会员管理系统

首页

用户管理

订单查询

会员等级管理

满减券管理

商品管理

订单一览

起始日期 - 截止日期

输入订单号或会员卡号

日期	订单号	会员卡号	订单金额	获得积分	使用满减券	操作
2025-09-04T08:15:19	207	10000006	¥127.00	22	满5.0减5.0	查看详情
2025-09-03T14:56:14	206	10000006	¥1.20	0	无	查看详情
2025-09-03T14:54:43	205	10000006	¥15.00	2	无	查看详情
2025-09-03T14:53:39	204	10000006	¥199.20	33	无	查看详情
2025-08-31T16:15:26	203	10000045	¥282.00	33	满100.0减50.0	查看详情
2025-08-31T16:15:15	202	10000045	¥25.00	2	无	查看详情
2025-08-31T16:15:14	201	10000045	¥20.00	2	无	查看详情
2025-08-31T16:15:09	200	10000045	¥45.00	4	无	查看详情
2025-08-31T16:15:07	199	10000045	¥6.00	0	无	查看详情
2025-08-31T16:15:05	198	10000045	¥50.00	5	无	查看详情
2025-08-31T16:15:00	197	10000045	¥10.00	1	无	查看详情
2025-08-31T16:14:57	196	10000045	¥24.00	2	无	查看详情

超市会员管理系统

首页

用户管理

订单查询

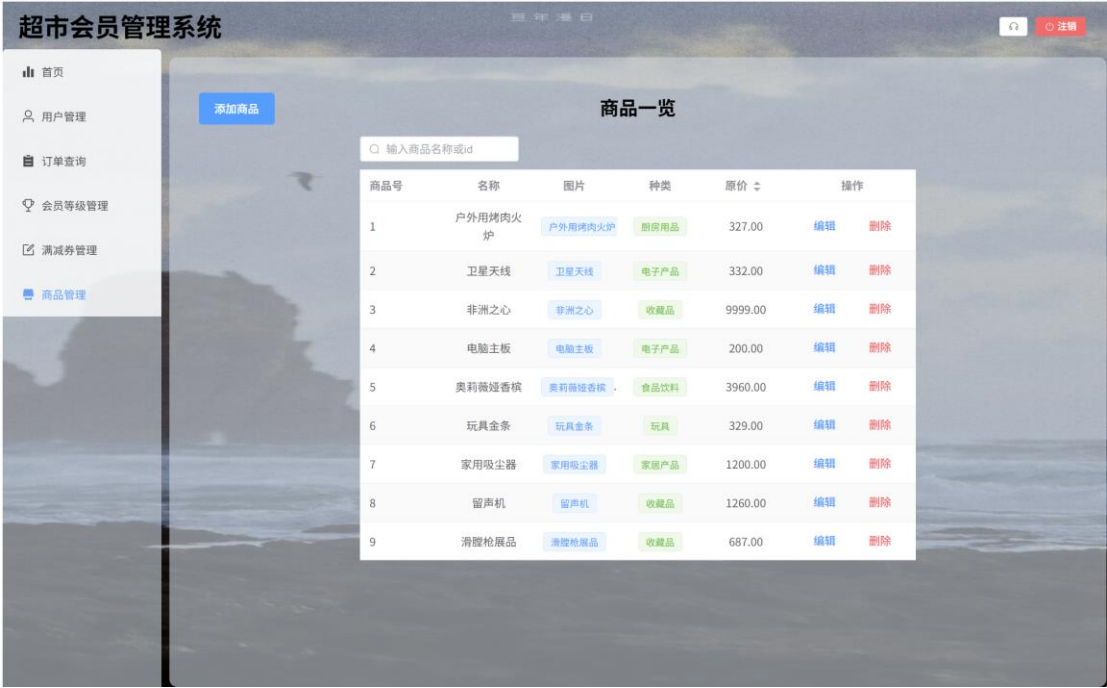
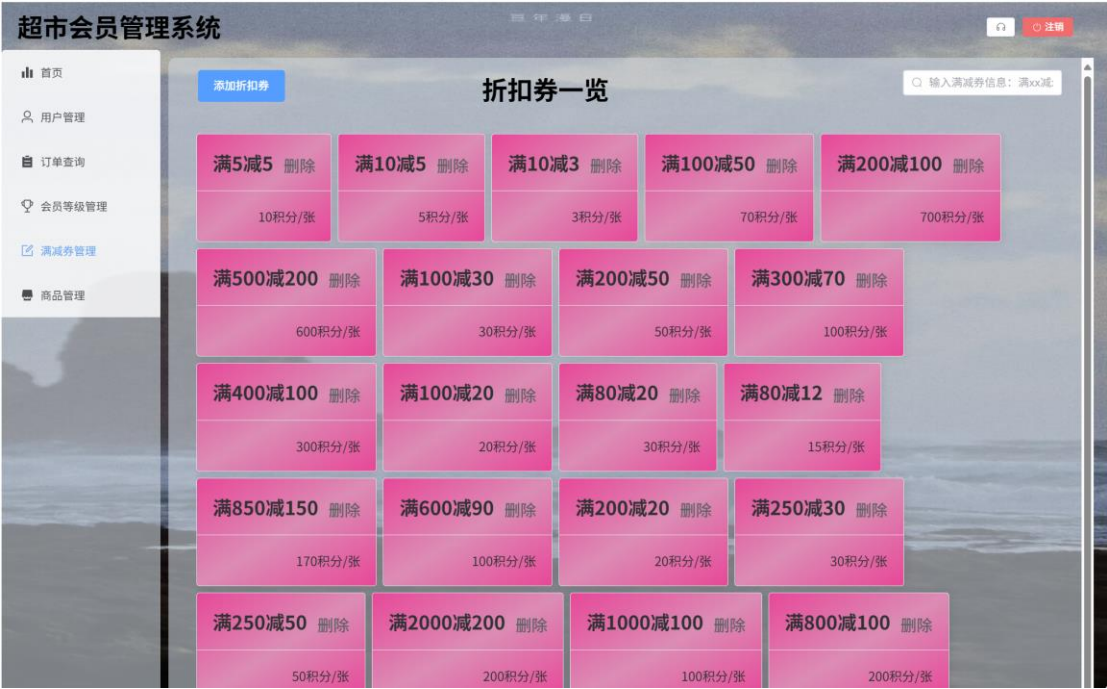
会员等级管理

满减券管理

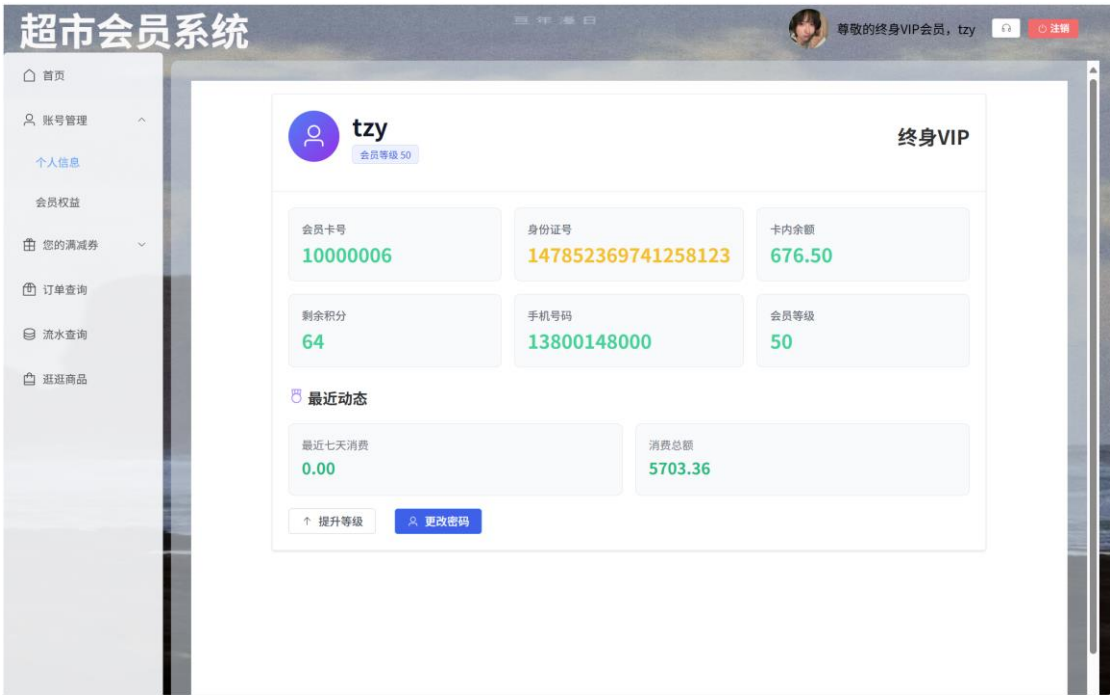
商品管理

会员等级及权益一览

<div>新客会员</div> <div>初始金额为0元 结算折扣为1折 购买享受积分回馈</div> <div>编辑</div>	<div>常客会员</div> <div>初始金额为300元 结算折扣为0.98折 购买享受积分回馈</div> <div>编辑</div>
<div>铜卡会员</div> <div>初始金额为500元 结算折扣为0.97折 购买享受积分回馈</div> <div>编辑</div>	<div>铜卡II级会员</div> <div>初始金额为1000元 结算折扣为0.96折 购买享受积分回馈</div> <div>编辑</div>
<div>银卡会员</div> <div>初始金额为1500元 结算折扣为0.95折 购买享受积分回馈</div> <div>编辑</div>	<div>银卡II级会员</div> <div>初始金额为2000元 结算折扣为0.94折 购买享受积分回馈</div> <div>编辑</div>
<div>银卡III级会员</div> <div>初始金额为2500元 结算折扣为0.93折 购买享受积分回馈</div> <div>编辑</div>	<div>金卡会员</div> <div>初始金额为3000元 结算折扣为0.92折 购买享受积分回馈</div> <div>编辑</div>
<div>金卡II级会员</div> <div>初始金额为3500元 结算折扣为0.91折 购买享受积分回馈</div> <div>编辑</div>	<div>金卡III级会员</div> <div>初始金额为4000元 结算折扣为0.9折 购买享受积分回馈</div> <div>编辑</div>



3. 用户界面



超市会员系统

三年 零 月 零 日

尊敬的终身VIP会员，tzy

注销

首页

账号管理

个人信息

会员权益

您的满减券

订单查询

流水查询

逛逛商品

您的会员等级

终身VIP会员

结算折扣为6.0折

购买享受积分回馈

新客会员

初始金额为0元

不能享受结算折扣

购买享受积分回馈

常客会员

初始金额为300元

结算折扣为9.8折

购买享受积分回馈

铜卡会员

初始金额为500元

结算折扣为9.7折

购买享受积分回馈

铜卡II级会员

初始金额为1000元

结算折扣为9.6折

购买享受积分回馈

银卡会员

初始金额为1500元

结算折扣为9.5折

购买享受积分回馈

银卡II级会员

初始金额为2000元

结算折扣为9.4折

购买享受积分回馈

超市会员系统

三年 零 月 零 日

尊敬的终身VIP会员，tzy

注销

首页

账号管理

个人信息

会员权益

您的满减券

查看满减券

兑换满减券

订单查询

流水查询

逛逛商品

5元折扣券

使用条件:满5元起

6张

5元折扣券

使用条件:满10元起

17张

3元折扣券

使用条件:满10元起

12张

70元折扣券

使用条件:满300元起

2张

30元折扣券

使用条件:满100元起

8张

50元折扣券

使用条件:满250元起

4张

12元折扣券

使用条件:满80元起

2张

50元折扣券

使用条件:满100元起

4张

20元折扣券

使用条件:满80元起

1张

20元折扣券

使用条件:满200元起

1张

超市会员系统

33 44 55 66

尊敬的终身VIP会员，tzy

注销

首页

账号管理

个人信息

会员权益

您的满减券

查看满减券

兑换满减券

订单查询

流水查询

逛逛商品

您的积分：64

折扣券只能使用在商品自身价格满足使用条件的情况下

输入满减券信息：满xx减xx

5元满减券

使用条件:满5元起

10积分/张

立即兑换

5元满减券

使用条件:满10元起

5积分/张

立即兑换

3元满减券

使用条件:满10元起

3积分/张

立即兑换

50元满减券

使用条件:满100元起

70积分/张

立即兑换

100元满减券

使用条件:满200元起

700积分/张

立即兑换

200元满减券

使用条件:满500元起

600积分/张

立即兑换

30元满减券

使用条件:满100元起

30积分/张

立即兑换

50元满减券

使用条件:满200元起

50积分/张

立即兑换

超市会员系统

33 44 55 66

尊敬的终身VIP会员，tzy

注销

首页

账号管理

个人信息

会员权益

您的满减券

查看满减券

兑换满减券

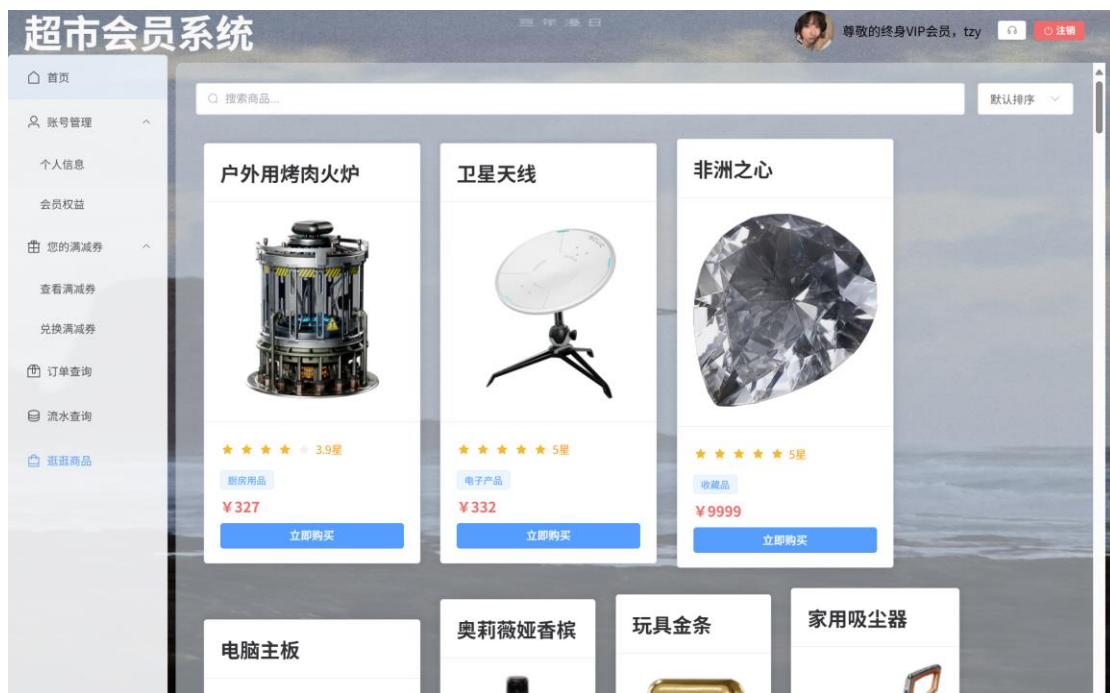
订单查询

流水查询

逛逛商品

订单详情

日期	订单号	订单金额	获得积分	使用满减券	操作
2025-09-04T08:15:19	207	¥127.00	22 积分	满5.0减5.0	查看详情
2025-09-03T14:56:14	206	¥1.20	0 积分	无	查看详情
2025-09-03T14:54:43	205	¥15.00	2 积分	无	查看详情
2025-09-03T14:53:39	204	¥199.20	33 积分	无	查看详情
2025-08-31T16:00:11	90	¥348.00	58 积分	无	查看详情
2025-08-31T15:59:08	89	¥15.00	2 积分	无	查看详情
2025-08-31T15:59:06	88	¥6.00	1 积分	无	查看详情
2025-08-31T15:59:03	87	¥3.59	0 积分	无	查看详情
2025-08-31T15:59:00	86	¥14.39	2 积分	无	查看详情
2025-08-31T15:58:58	85	¥64.80	10 积分	无	查看详情
2025-08-31T15:58:56	84	¥2.40	0 积分	无	查看详情



7.3 事务设计

7.3.1 新用户注册

保证不是已注册的客户，保证身份证号和手机号的正确性。

```

@PostMapping("/login")
public Result login(@RequestBody Map<String, Object> parameters) {
    //遍历 map
    parameters.forEach((key, value) -> Log.info("Key: " + key + "
Value: " + value));
    Map<String, Object> map = userService.login(
        Integer.valueOf((String) parameters.get("account")),
        (String) parameters.get("password"));
    return new Result(map);
}

```

```

@Override
public Map<String, Integer> register(User user) {
    save(user);
    Log.info("Register new user: " + user);
    Map<String, Integer> map = new HashMap<>(1);
    map.put("account", user.getAccount());
    return map;
}

```

7.3.2 用户修改密码和充值

修改密码

```

@PostMapping("/change_password")
public Result changePassword(@RequestBody Map<String, String>
parameters) {
    Integer account =

```

```

JwtUtils. getUserAccount(request.getHeader(JwtUtils. AUTH_HEADER_KEY));
    return userService.changePassword(
        account,
        parameters.get("oldPassword"),
        parameters.get("newPassword")
    )? Result. success() : new Result("原密码输入错误");
}

```

充值

@Override

@Transactional//加入 @transactional 注解，抛出异常之后，事务会自动回滚，数据不会插入到数据库中。

```

public boolean recharge(Integer account, Double amount) {
    //MyBatis-Plus 查询构造器，用于构建 SQL 条件
    QueryWrapper<User> queryWrapper = new QueryWrapper<>();
    //按 account 查询 相当于 WHERE account = #{account}, 指定只查询
balance 字段
    queryWrapper.eq("account", account).select("balance");
    List<Map<String, Object>> list = listMaps(queryWrapper);
    Double balance = (Double) list.get(0).get("balance");
    //MyBatis-Plus 更新构造器
    UpdateWrapper<User> updateWrapper = new UpdateWrapper<>();
    //更新条件 WHERE account = #{account}, 设置新余额 SET balance =
#{当前余额 + 充值金额}
    updateWrapper.eq("account", account).set("balance", amount +
balance);
    if (update(updateWrapper)) {
        Charge charge = new Charge();

```

```

        charge.setUserId(account);
        charge.setAmount(amount);
        charge.setTime(LocalDateTime.now());
        log.info("charge:" + charge);
        return chargeService.save(charge);
    } else {
        return false;
    }
}

```

7.3.3 购买商品

用户查看商品的基本信息，并可以购买商品

```

@Override
@Transactional
public Map<String, Object> checkout(Integer account, OrderDTO
orderDTO) {
    double finalPrice;          // 最终支付价格
    String couponInfo = null;    // 优惠券信息（格式：满减条件/减免金
    额）
    double price = 0.00D;        // 商品初始总价

    CommodityDTO commodityDTO =
orderDTO.getCommodityDTOList().get(0);
    double commodityPrice =
commodityService.getPrice(commodityDTO.getCommodityId());
    price = commodityPrice * commodityDTO.getNumber(); // 直接计算总
    价

    // 计算订单返还的积分

```

```
int points = (int) (price * REWARD_POINTS_RETURN_RATE);
// 获取用户信息 用户等级对应的折扣
User user = userService.getById(account);
double discount =
membershipLevelService.getDiscount(user.getLevel());

if (orderDTO.getCouponType() != null) {
    // 若使用折扣券，查询使用的折扣券
    Coupon coupon =
couponService.getById(orderDTO.getCouponType());
    if (price > coupon.getLowestAmount()) {
        finalPrice = price * discount -
coupon.getReductionAmount();
        // 减少会员对应的满减券
        if (!userCouponService.useCoupon(account,
coupon.getType())) {
            throw new BaseException("结算失败，会员不持有该满减券
");
        }
        couponInfo = coupon.getLowestAmount() + "/" +
coupon.getReductionAmount();
    } else {
        throw new BaseException("结算失败，未能达成使用满减券条件
");
    }
} else {
    finalPrice = price * discount;
}

//防止极端情况（如优惠券减免金额过大）导致负价格。
```

```
finalPrice = finalPrice >= 0 ? finalPrice : 0;
// 减少会员余额
if (!userService.deductBalance(account, finalPrice)) {
    throw new BaseException("结算失败, 会员卡内余额不足");
}
// 赠送会员积分
if (!userService.addPoints(account, points)) {
    throw new BaseException("结算失败, 赠送会员积分失败");
}
// order 表插入信息
UserOrder userOrder = new UserOrder();
userOrder.setUserId(account);
userOrder.setPaymentAmount(finalPrice);
userOrder.setRewardPoints(points);
userOrder.setTime(LocalDateTime.now());
userOrder.setCouponInfo(couponInfo);
if (!save(userOrder)) {
    throw new BaseException("结算失败, 订单记录存储失败");
}
// 插入 order_commodity 表信息
OrderCommodity orderCommodity = new OrderCommodity();
orderCommodity.setOrderId(userOrder.getId());
orderCommodity.setCommodityId(commodityDTO.getCommodityId());
orderCommodity.setQuantity(commodityDTO.getNumber());
orderCommodity.setOriginalPrice(price);
orderCommodity.setActualPrice(finalPrice);

if (!orderCommodityService.save(orderCommodity)) {
    throw new BaseException("结算失败, 订单商品记录存储失败");
}
```



```

    }

    Map<String, Object> returnMap = new HashMap<>(2);
    returnMap.put("finalPrice", finalPrice);
    returnMap.put("points", points);
    return returnMap;
}

```

7.3.4 兑换满减券

用户获得积分后，可以兑换满减券

```

@Override
@Transactional
public boolean redeem(Integer account, Integer couponType, Integer
number) {
    // 查询会员积分
    QueryWrapper<User> queryWrapper = new QueryWrapper<>();
    queryWrapper.eq("account", account).select("points");
    Map<String, Object> conditions = getMap(queryWrapper);
    Integer points = (Integer) conditions.get("points");

    // 查询兑换的折扣券所需要的积分并且计算出总共需要的积分
    QueryWrapper<Coupon> couponQueryWrapper = new QueryWrapper<>();
    couponQueryWrapper.eq("type", couponType).select("cost");
    conditions = couponService.getMap(couponQueryWrapper);
    Integer costPoints = (Integer) conditions.get("cost") * number;

    // 判断会员的积分是否足够
    if (points >= costPoints) {
        // 当足够兑换时，先在 user_coupon 中查找会员对应满减券的数
        量，然后加上兑换的数量，
    }
}

```

```

        // 若没有记录则新建一条
        UpdateWrapper<UserCoupon> userCouponUpdateWrapper = new
UpdateWrapper<>();
        userCouponUpdateWrapper.eq("user_id", account)
            .eq("coupon_type", couponType).setSql("quantity =
quantity + " + number);
        // 若更新失败, 则将 entity 插入。以下为 entity
        if (!userCouponService.update(userCouponUpdateWrapper)) {
            UserCoupon userCoupon = new UserCoupon();
            userCoupon.setUserId(account);
            userCoupon.setCouponType(couponType);
            userCoupon.setQuantity(number);
            if (!userCouponService.save(userCoupon)) {
                throw new BaseException("兑换失败: 增加会员满减券数量
失败");
            }
        }
    }

    //扣除对应积分
    UpdateWrapper<User> updateWrapper = new UpdateWrapper<>();
    updateWrapper.eq("account", account).set("points", points -
costPoints);
    // 向 redeem 表中插入兑换记录
    if (update(updateWrapper)) {
        Redeem redeem = new Redeem();
        redeem.setUserId(account);
        redeem.setCouponType(couponType);
        redeem.setNumber(number);
        redeem.setValue(costPoints);
    }
}

```

```

        redeem.setTime(LocalDateTime.now());
        if (redeemService.save(redeem)) {
            return true;
        } else {
            throw new BaseException("兑换失败：兑换记录存储失败");
        }
    } else {
        throw new BaseException("兑换失败：扣除会员积分失败");
    }
}

return false;
}

```

7.3.5 商品修改

管理员可以修改火车票的基本信息

```

async function submitAddCommodity(formEl) {

    if (!formEl) return

    await formEl.validate((valid) => {

        if (valid) {

            axiosT.post('/commodity/add', {

                id: null,

                name: addCommodityData.value.name,

                price: addCommodityData.value.price,

                rating: addCommodityData.value.rating,

```

```
        description: addCommodityData.value.description,

        image: addCommodityData.value.image,

        category: addCommodityData.value.category
    }, (data) => {

        commoditiesList.value.push({

            id: data,

            name: addCommodityData.value.name,

            rating: addCommodityData.value.rating,

            price: addCommodityData.value.price,

            image: addCommodityData.value.image,

            description: addCommodityData.value.description,

            category: addCommodityData.value.category

        })

        search()

        ElMessage.success(' 添加商品成功')

    })

    addCommodityVisible.value = false

} else {

    ElMessage({

        showClose: true,

        message: ' 输入的信息不符合要求',
```

```

        type: 'error'

    })

}

})

}

```

7.3.6 充值金额统计

管理员可以查看用户的充值总额

```

@GetMapping("/administrator/totalAmount")
public Result getTotalAmount() {
    checkIdentity();
    double totalAmount = 0;
    List<Charge> list = chargeService.list();
    for (Charge charge : list) {
        totalAmount += charge.getAmount();
    }
    return new Result(totalAmount);
}

```

7.3.7 消费金额统计

管理员可以查看用户的消费总额

```

@GetMapping("/administrator/totalAmount")
public Result getTotalAmount() {
    checkIdentity();

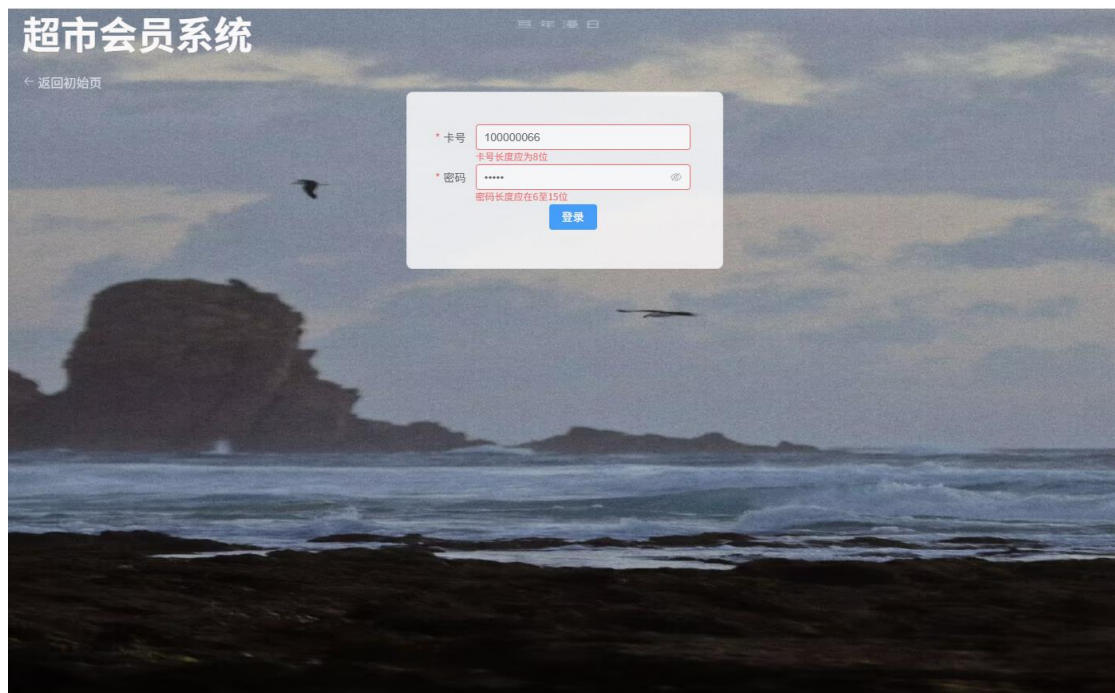
```

```
double totalAmount = 0;
List<UserOrder> list = userOrderService.list();
for (UserOrder userOrder : list) {
    totalAmount += userOrder.getPaymentAmount();
}
return new Result(totalAmount);
}
```

八、测试和运行

8.1 登录

登录一切正常，只要用户名、密码正确即可，任意一个不对都会报错



8.2 管理员

管理员登陆后，定义的各项功能均可实现

修改用户信息/查询用户最近的动态

卡号	姓名	手机号	身份证号	余额	积分	会员等级	操作
10000052	njbwe	13184225581	147*****3	888.00	0	1	查看动态 修改信息
10000051	mknqdw	13184225523	147*****4	113.00	10	2	查看动态 修改信息
10000050	mkwe	13184225584	147*****4		0	1	查看动态 修改信息
10000049	nkonkerg	13800148000	147*****4		0	1	查看动态 修改信息
10000048	noaedi	13800148000	147*****4		0	1	查看动态 修改信息
10000047	nuorg	13800148000	123*****6		0	1	查看动态 修改信息
10000046	mkegreg	13800148000	123*****5	79.00	5651	1	查看动态 修改信息
10000045	bjwef	13800148000	147*****7	59.00	815	1	查看动态 修改信息

查看用户订单（可以按照用户卡号查询或者日期查询）

超市会员管理系统

山 首页

🔍 用户管理

📄 订单查询

👤 会员等级管理

📄 满减券管理

📄 商品管理

订单一览

📅 起始日期

📅 截止日期

🔍 输入订单号或会员卡号

日期	订单号	会员卡号	订单金额	获得积分	使用满减券	操作
2025-09-04T08:15:19	207	10000006	¥127.00	22	满5.0减5.0	查看详情
2025-09-03T14:56:14	206	10000006	¥1.20	0	无	查看详情
2025-09-03T14:54:43	205	10000006	¥15.00	2	无	查看详情
2025-09-03T14:53:39	204	10000006	¥199.20	33	无	查看详情
2025-08-31T16:15:26	203	10000045	¥282.00	33	满100.0减50.0	查看详情
2025-08-31T16:15:15	202	10000045	¥25.00	2	无	查看详情
2025-08-31T16:15:14	201	10000045	¥20.00	2	无	查看详情
2025-08-31T16:15:09	200	10000045	¥45.00	4	无	查看详情
2025-08-31T16:15:07	199	10000045	¥6.00	0	无	查看详情
2025-08-31T16:15:05	198	10000045	¥50.00	5	无	查看详情
2025-08-31T16:15:00	197	10000045	¥10.00	1	无	查看详情
2025-08-31T16:14:57	196	10000045	¥24.00	2	无	查看详情
2025-08-31T16:14:57	195	10000045	¥2.00	0	无	查看详情

管理会员等级

超市会员管理系统

山 首页

🔍 用户管理

📄 订单查询

👤 会员等级管理

📄 满减券管理

📄 商品管理

会员等级及权益一览

新客会员

初始金额为0元
结算折扣为1折
购买享受积分回馈

编辑

常客会员

初始金额为300元
结算折扣为0.98折
购买享受积分回馈

编辑

铜卡会员

初始金额为500元
结算折扣为0.97折
购买享受积分回馈

编辑

铜卡II级会员

初始金额为1000元
结算折扣为0.96折
购买享受积分回馈

编辑

银卡会员

初始金额为1500元
结算折扣为0.95折
购买享受积分回馈

编辑

银卡II级会员

初始金额为2000元
结算折扣为0.94折
购买享受积分回馈

编辑

银卡III级会员

初始金额为2500元
结算折扣为0.93折
购买享受积分回馈

编辑

金卡会员

初始金额为3000元
结算折扣为0.92折
购买享受积分回馈

编辑

金卡II级会员

初始金额为3500元
结算折扣为0.91折
购买享受积分回馈

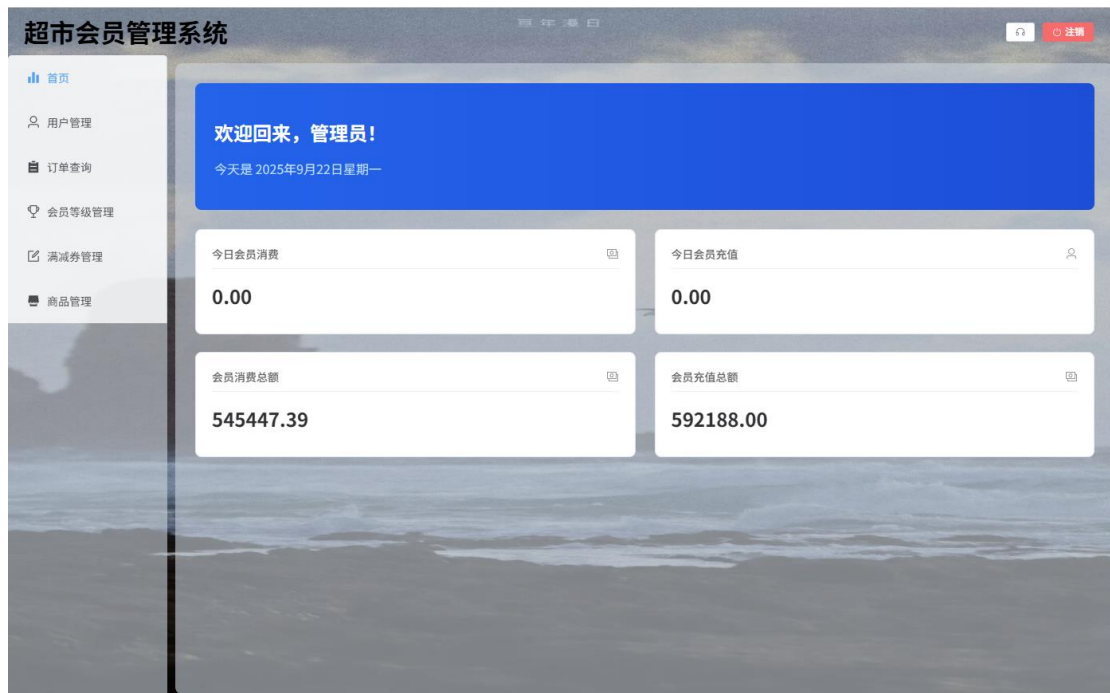
编辑

金卡III级会员

初始金额为4000元
结算折扣为0.9折
购买享受积分回馈

编辑

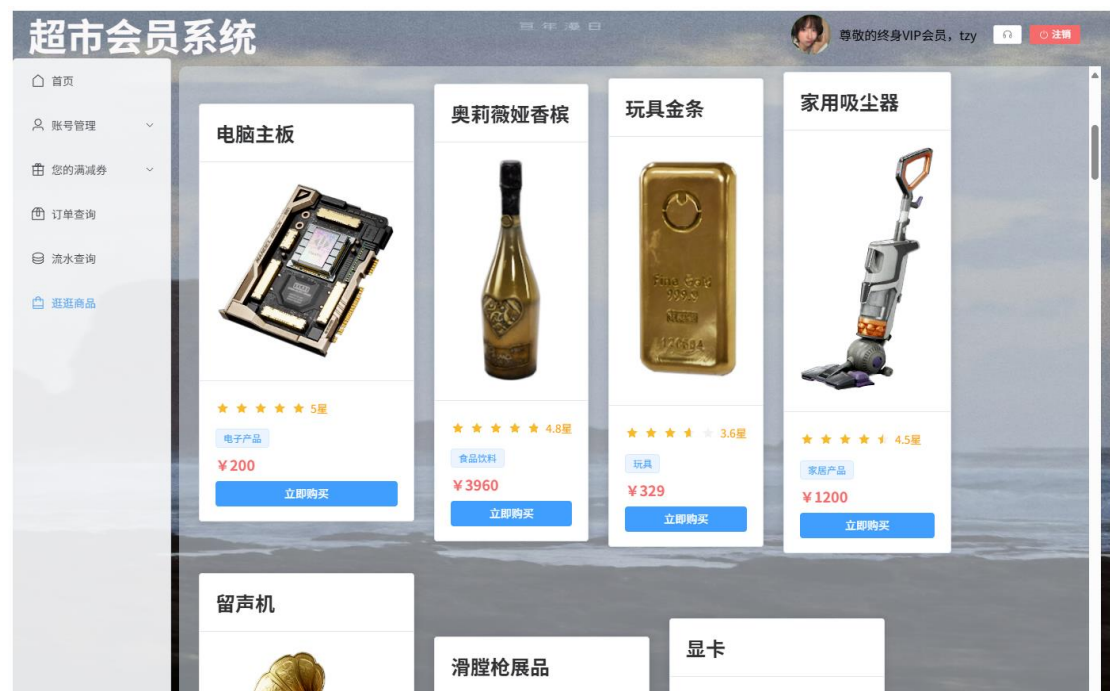
销售金额/充值金额统计



8.3 用户

各项功能实现正常

商品购买





订单查询

超市会员系统

🏠 首页

👤 账号管理

👛 您的满减券

📄 订单查询

💰 流水查询

🛒 逛逛商品

👤 尊敬的终身VIP会员，tzy

🔔 消息

🔑 注销

订单详情

日期	订单号	订单金额	获得积分	使用满减券	操作
2025-09-04T08:15:19	207	¥127.00	22 积分	满5.0减5.0	查看详情
2025-09-03T14:56:14	206	¥1.20	0 积分	无	查看详情
2025-09-03T14:54:43	205	¥15.00	2 积分	无	查看详情
2025-09-03T14:53:39	204	¥199.20	33 积分	无	查看详情
2025-08-31T16:00:11	90	¥348.00	58 积分	无	查看详情
2025-08-31T15:59:08	89	¥15.00	2 积分	无	查看详情
2025-08-31T15:59:06	88	¥6.00	1 积分	无	查看详情
2025-08-31T15:59:03	87	¥3.59	0 积分	无	查看详情
2025-08-31T15:59:00	86	¥14.39	2 积分	无	查看详情
2025-08-31T15:58:58	85	¥64.80	10 积分	无	查看详情
2025-08-31T15:58:56	84	¥2.40	0 积分	无	查看详情

流水查询

超市会员系统

2025年9月10日

尊敬的终身VIP会员，tzy

注册

首页

账号管理

您的满减券

订单查询

流水查询

逛逛商品

流水查询

日期	余额变动	积分变动	事项单号	事项备注
2025-09-04T08:15:19	-127.00	+22	207	购物订单
2025-09-03T14:56:14	-1.20	+0	206	购物订单
2025-09-03T14:54:43	-15.00	+2	205	购物订单
2025-09-03T14:53:39	-199.20	+33	204	购物订单
2025-08-31T16:00:45	-	-20	218	满减券兑换
2025-08-31T16:00:42	-	-30	217	满减券兑换
2025-08-31T16:00:40	-	-15	216	满减券兑换

Total 817/page12Go to1

查看已有的满减券

超市会员系统

2025年9月10日

尊敬的终身VIP会员，tzy

注册

首页

账号管理

个人信息

会员权益

您的满减券

查看满减券

兑换满减券

订单查询

流水查询

逛逛商品

5元折扣券

使用条件:满5元起

6张

5元折扣券

使用条件:满10元起

17张

3元折扣券

使用条件:满10元起

12张

70元折扣券

使用条件:满300元起

2张

30元折扣券

使用条件:满100元起

8张

50元折扣券

使用条件:满250元起

4张

12元折扣券

使用条件:满80元起

2张

50元折扣券

使用条件:满100元起

4张

20元折扣券

使用条件:满80元起

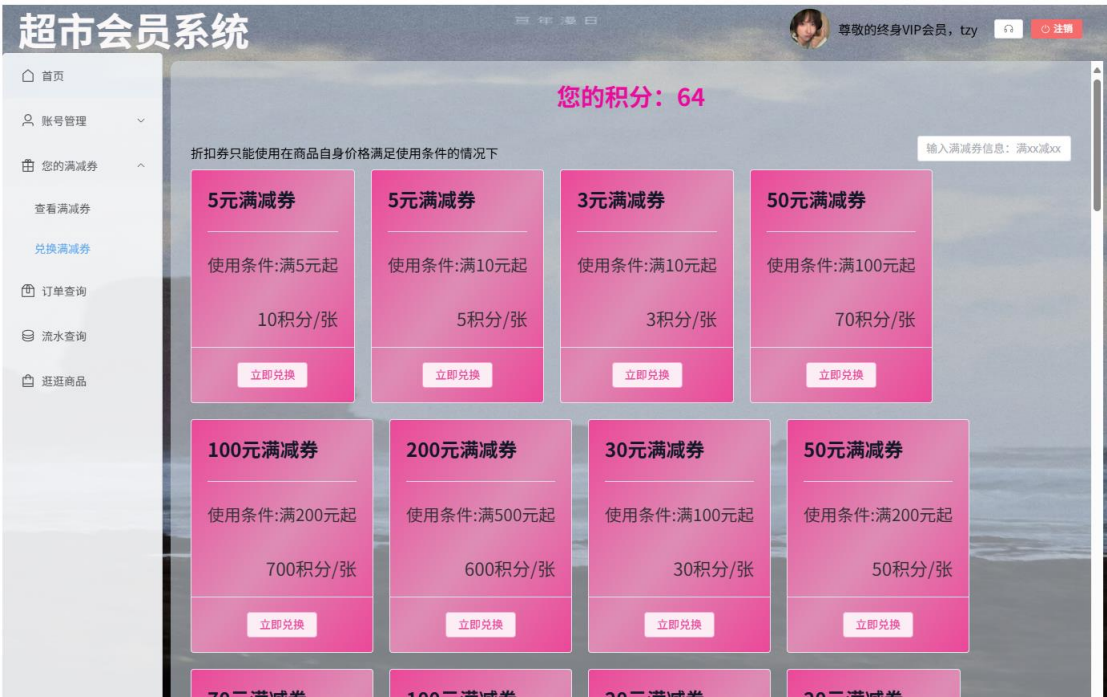
1张

20元折扣券

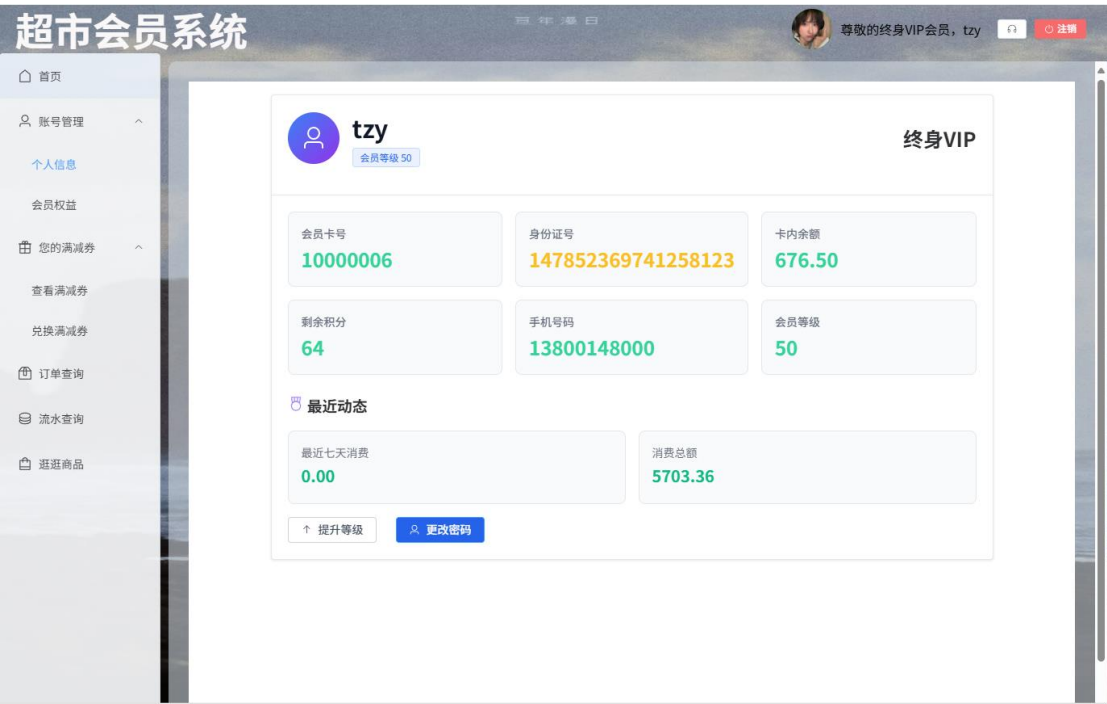
使用条件:满200元起

1张

兑换满减券



管理个人信息



九、总结

在完成超市会员管理系统的数据库课程设计过程中，我经历了一次从理论认知到实践应用的重要蜕变。在这个过程中，我不仅锻炼了技术能力，更培养了独立解决问题的系统思维和工程化意识。

在数据库设计阶段，我深刻体会到了理论知识与实践应用的差距。教科书上的规范化理论看似简单明了，但在实际设计中却需要综合考虑性能、扩展性和业务需求的多重约束。在开发过程中遇到的种种困难让我明白，数据库设计不是简单的理论套用，而是需要在理想与现实之间找到最佳平衡点。

SQL 编程方面，我使用了 mybatis-plus 这一持久层，把复杂的 SQL 语句编写变成了对不同 Wrapper 对象的方法调用。这个过程让我对数据库的事务处理和并发控制有了更深刻的理解。

作为个人开发项目，我不得不独自承担从需求分析到系统部署的全流程工作。这个过程培养了我的系统思维能力。在需求分析阶段，我学会了如何从模糊的业务描述中提取出精确的数据需求；在架构设计阶段，我考虑了系统的可扩展性和维护性，为每个表都添加了创建时间和更新时间字段，为可能的功能扩展预留了接口；在测试阶段，我编写了全面的测试用例，包括正常流程测试、边界条件测试和异常情况测试。

最让我受益匪浅的是故障排查能力的提升。在开发过程中，我遇到了特别棘手的问题，调试过程虽然花费了大量时间，但让我对数据库的并发机制有了前所未有的深刻认识。

这次个人开发经历让我初步建立了软件工程的系统化思维。我学会了使用版本控制来管理数据库变更脚本，使用文档来记录设计决策，使用注释来保证代码的可维护性。我意识到良好的工程实践不是负担，而是提高开发效率和质量的重要保障。

通过这次课设，我不仅掌握了数据库开发的技术技能，更重要的是培养了独立解决问题的能力 and 持续学习的习惯。每个遇到的问题都成为我深入学习的机会，每次调试的过程都加深了我对系统运行机制的理解。我认识到数据库技术不仅是存储数据的工具，更是业务逻辑的核心载体。良好的数据库设计能够提高系统的稳定性和可维护性，而糟糕的设计则可能导致灾难性的后果。这种认识将指导我在未来的开发工作中更加重视数据层的设计和优化。回首整个开发过程，从最初的需求迷茫到中期的技术挑战，再到最后的系统完善，我收获的不仅仅是一个可以运行的超市会员管理系统，更是一套完整的数据库开发方法论和解决问题的能力体系。这段经历让我确信，只有将理论知识与实践相结

合，才能真正掌握数据库技术的精髓。我相信这次课设积累的经验和教训，将成为我未来职业发展的宝贵财富。