

二手交易平台

(线上市场) 概要设计

文件状态: 正式发布

文件标识: 线上市场

当前版本: v1.2

作者:

姓名: 吴昊、张羿凡、...

学号:

团队: 啊对对队

完成日期: 2021/10/20

版本更新信息:

v0.5 发布说明

v1.0 补充信息

v1.2 增加详细说明

1 引言

1.1 使用人员:

用户
开发人员
项目经理
营销人员
测试人员
文档编写人员

1.2 编写目的:

软件概要设计是从总体上把握系统设计框架, 他包括模块划分、处理流程和接口设计, 概要设计对上述内容作了总体描述, 体现了用户需求与应用系统实现之间的关系, 在设计过程中起到了指导作用。本阶段已在系统的需求分析的基础上, 对二手交易平台做概要设计。主要解决了实现该系统程序模块设计问题。包括如何把该系统划分成若干个模块、决定各个模块之间的接口、模块之间的信息, 以及数据结构、模块结构的设计等。在以下的概要设计报告中将对在本阶段中对系统所做设计进行详细的说明。在下一阶段的详细设计中, 程序设计员可参考此概要设计报告, 在概要设计交易平台所做的模块结构设计的基础上, 对系统进行详细设计。在以后的软件测试以及软件维护参考此说明书, 以便于了解在概要设计过程中所完成的各模块设计结构, 或在修改时找出在开发中的不足或错误。

待开发的软件系统的名称: 二手交易平台

项目名称: 二手交易平台

项目的任务提出者: 阿里巴巴集团控股有限公司

项目的任务开发者: 啊对对队

项目的用户: 有二手交易需求的用户

本文档的读者: 二手交易平台项目组

1.3 背景:

软件系统的名称: 二手交易平台

此项目由阿里巴巴发布招标、啊对对队接受委托开发, 面向全部需求二手交易的用户开发。系统由两部分组成: 基于阿里云的数据库服务器, 以及置于银行的数据库服务器。

1.4 定义与缩写

术语:

解释:

连接数据库使用的jar包

术语:

解释:

拦截所有外部请求

术语:

解释:

数据库增删改查操作

1.5 参考资料

- 《开发计划》
- 《需求规格说明书》

2 总体设计

2.1 需求规定:

输入项目:

1. 用户注册登录
2. 用户的销售商品
3. 商品的名称
4. 商品的数量
5. 商品的描述
6. 商品的价格

输出项目:

1. 销售订单
2. 详细商品列表
3. 物流数据

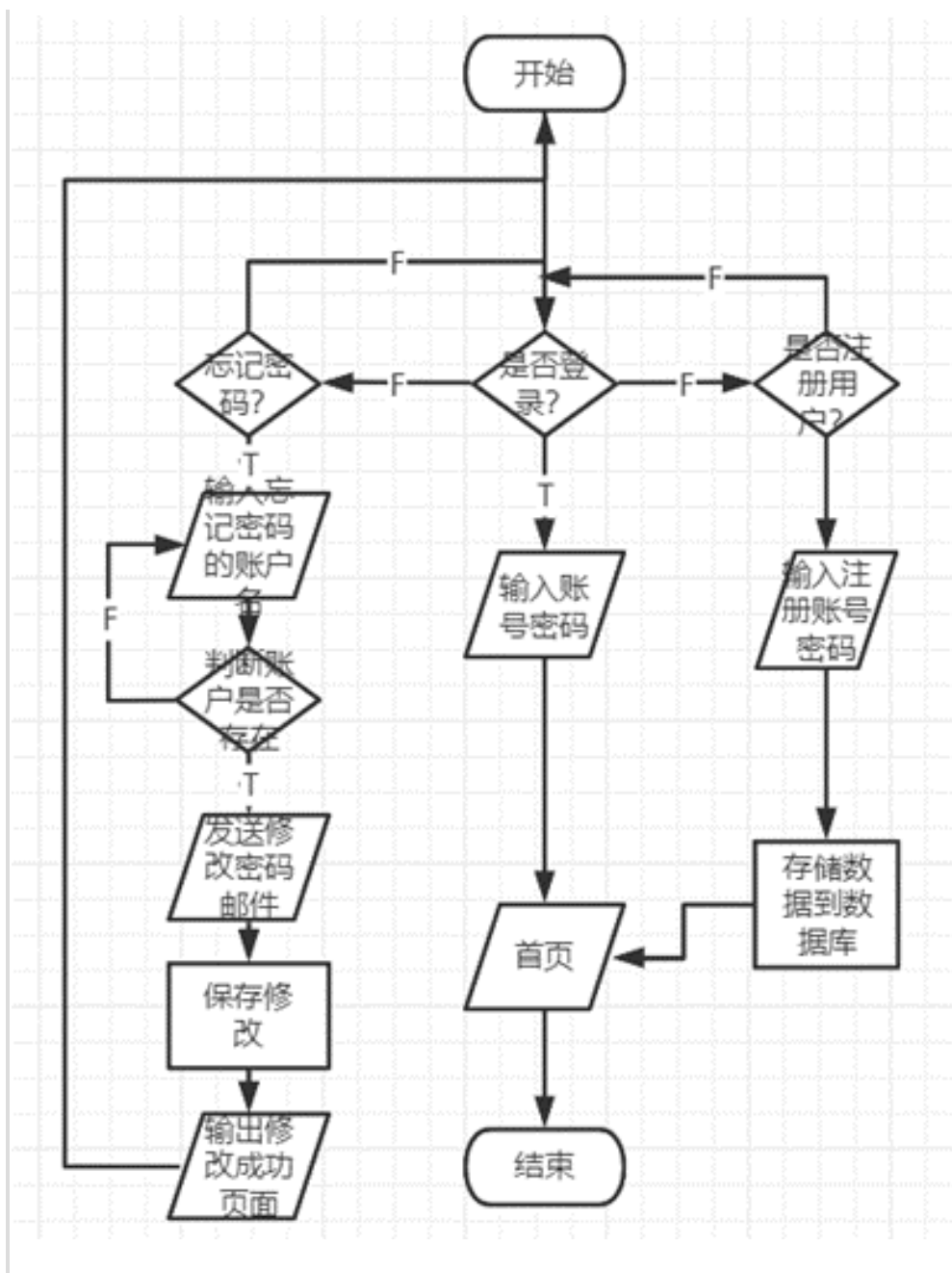
处理的功能新要求:

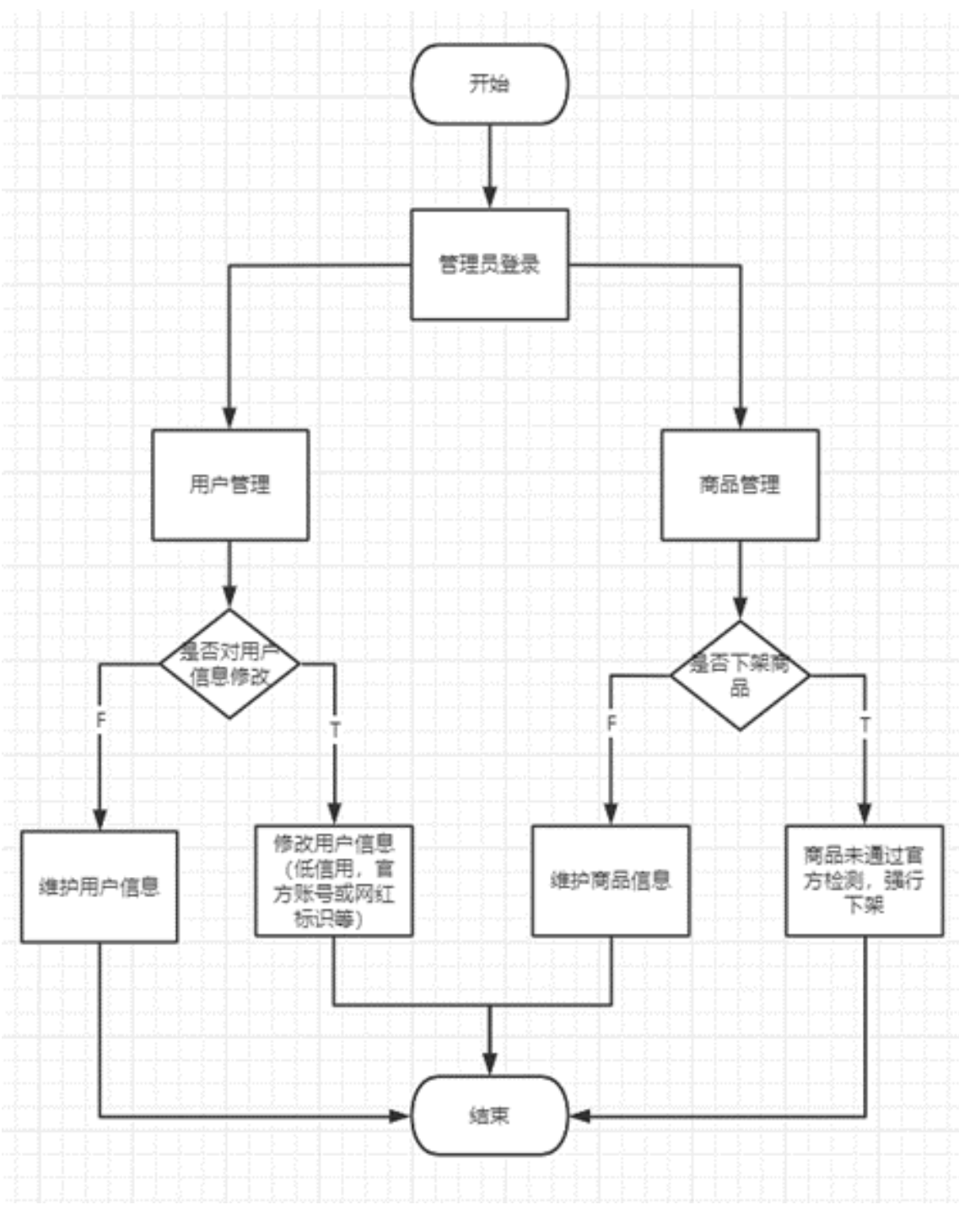
1. 对管理员用户输入的销售项目进行生成订单并执行销售
2. 刷新当前数据, 获取物流数据

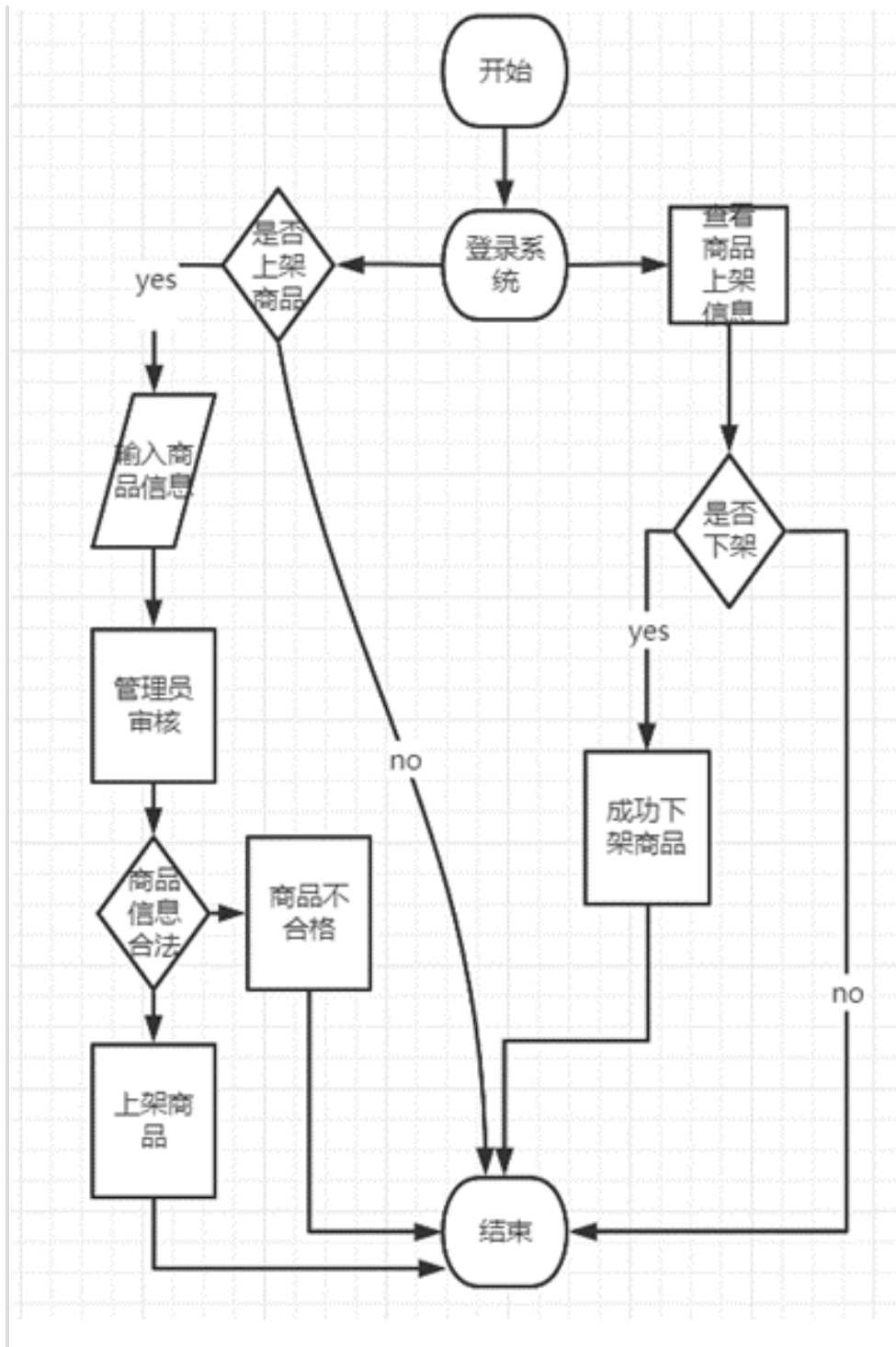
2.2 运行环境:

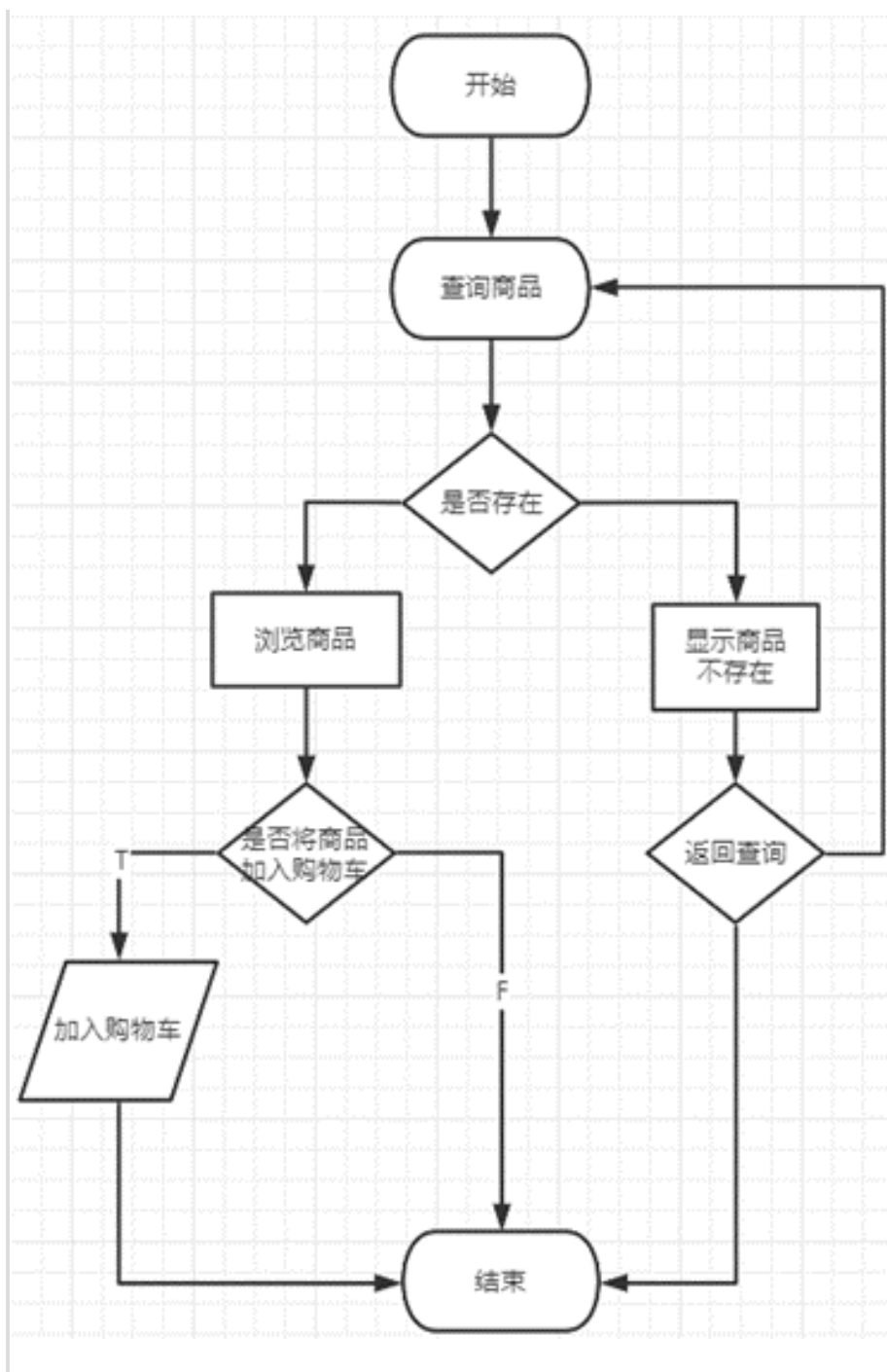
1. Android 3.1以上、Windows 7、Linux.
2. 主要联网才可进行正常的使用
3. 未登录用户可以浏览商品及其信息, 登录后才能使用其他功能

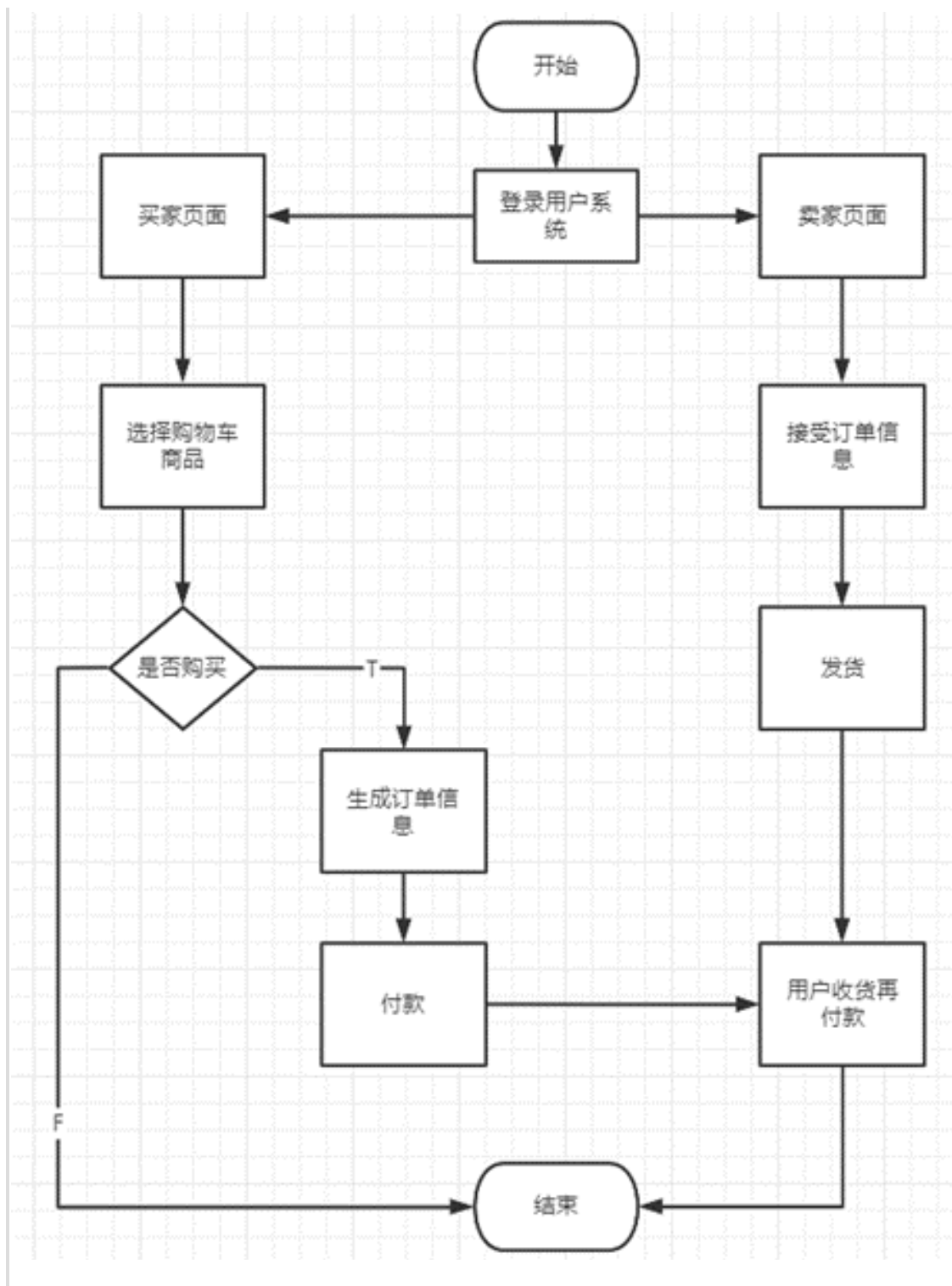
2.3 基本设计概念和处理流程:

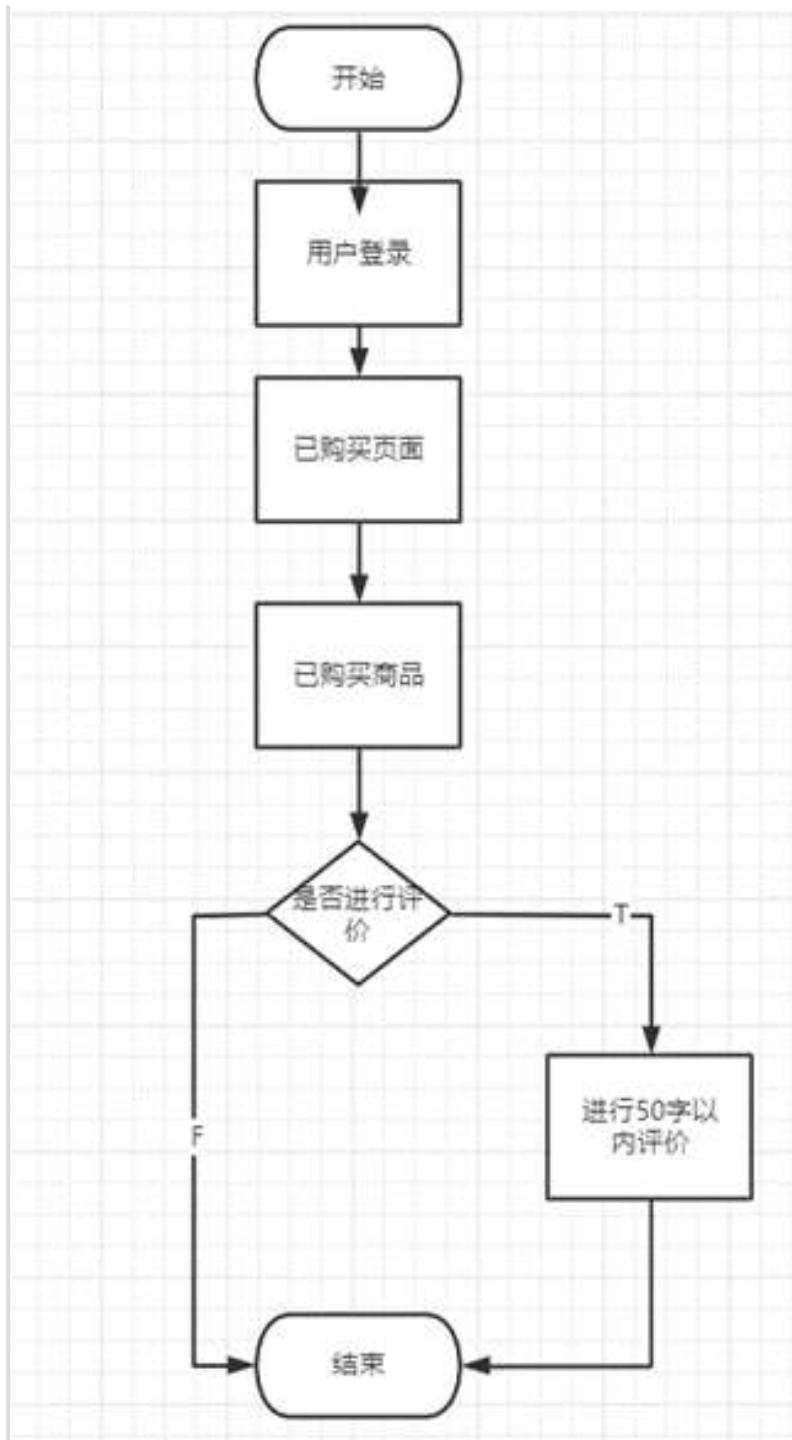












2.4 结构:

上架商品		√	√	
添加购物车		√	√	
订单执行		√	√	√

2.6 人工处理过程:

商品交易过程中买方卖方出现分歧, 需要客服介入调解

2.7 尚未问决的问题:

用户过多情况下并发使用该软件是否不会崩溃, 需要进行压力测试。
商品交易及订单生成是否能在短时间内完成, 没有过多延迟, 需要测试。

2.8 自由表述:

无

3 接口设计

3.1 用户接口:

1. 添加商品至用户购物车接口

"/addcar" POST

```
public void addgood(String id,String word,String shopid,String goodid,String name
price,String intro,String newo,String fenlei,String size,String yijia,String coun
amount,String imgUrl)
```

有库存则返回添加成功, 否则失败

2. 获取用户当前的所有购物车的商品接口

@RequestMapping(value = "/car",method = RequestMethod.POST)

```
public List<car> car(String id,String word)
```

返回所有当前用户的购物车商品信息

3. 从购物车中删除某个商品接口

@RequestMapping(value = "/car",method = RequestMethod.POST)

```
public List<car> car(String id,String word)
```

返回删除成功

4. 用户付款后的商品添加到已付款订单信息中接口

@RequestMapping(value = "/pay",method = RequestMethod.POST)

```
public void pay (String id,String word,String[] goodid,String phone,String uname,
address,String money)
```

无返回信息

5. 获取用户已付款的商品信息接口

@RequestMapping(value = "/payed",method = RequestMethod.POST)

```
public List<car> payed(String id,String word)
```

返回所有的当前用户已付款的商品列表信息

6. 用户商品退款接口

@RequestMapping(value = "/tuikuan",method = RequestMethod.POST)

```
public String tuikuan(String id,String word,String goodid,String time)
```

无返回信息

9. 获取商家退款订单

```
@RequestMapping(value = "/tui", method = RequestMethod.POST)
```

```
public List<car> tui(String id, String word)
```

无返回信息

10. 商家拒绝某商品的退款

```
@RequestMapping(value = "/jutui", method = RequestMethod.POST)
```

```
public void jutui(String id, String shopid, String word, String goodid)
```

无返回信息

11. 用户商家管理员登录接口

```
@RequestMapping(value = "/login", method = RequestMethod.POST)
```

```
public String login(String id, String word)
```

返回登录成功或者登录失败信息

12. 用户注册接口

```
@RequestMapping(value = "/insertZhuce", method = RequestMethod.POST)
```

```
public void insertZhuce(String id, String word, String name, String sex, String phone  
email, String city, String account)
```

返回注册成功信息

3.2 外部接口

硬件接口：

光纤

usb

电源

软件接口：

获取当前商家待发货所有商品信息

```
@RequestMapping(value = "/sendgood", method = RequestMethod.POST)
```

```
public List<car> sendgood(String id, String word)
```

返回当前商家所有的待发货的商品信息

支付接口，通过调用支付宝接口支付

3.3 内部接口

硬件接口：

光纤

局域网

软件接口：

1. 用户商家管理员登录接口

```
@RequestMapping(value = "/login", method = RequestMethod.POST)
```

```
public String login(String id, String word)
```

返回登录成功或者登录失败信息

所有的请求都要调用此接口来判断用户的权限，进而完成其对应的业务

2. jdbc

所有接口都得来通过此接口对数据库进行crud

4.1 运行模块组合：

- (1) 用户：登录模块+注册模块+商家注册模块+购物车模块
- (2) 主页：商品模块
- (3) 商家：管理商品模块+订单模块+退款模块
- (4) 管理员：登录模块+审核模块+商家管理模块

4.2 运行控制：

- (1) 在客户端登录后数据库验证账号密码，通过后进入主界面；
- (2) 新用户注册，提交用户名、密码、短信验证码；
- (3) 管理员通过注册请求，更新数据库；
- (4) 主界面显示当前所有在售的商品；
- (5) 用户购买商品，填写收货信息并付款；
- (6) 商家发货，买家确认收货，购买流程结束；
- (7) 商家申请上架商品，提交商品图片和信息审核；
- (8) 管理员审核商品上架，上架商品流程结束。

4.3 运行时间：

本系统的各个模块之间功能是相互联系的，故各模块占用各种资源的时间是累计的，所以运行的操作和网络的运行速度决定。

时间规定：用户点击按钮，服务器综合响应时间不超过2s、数据库查询不超过1s、数据处理不超过1s。

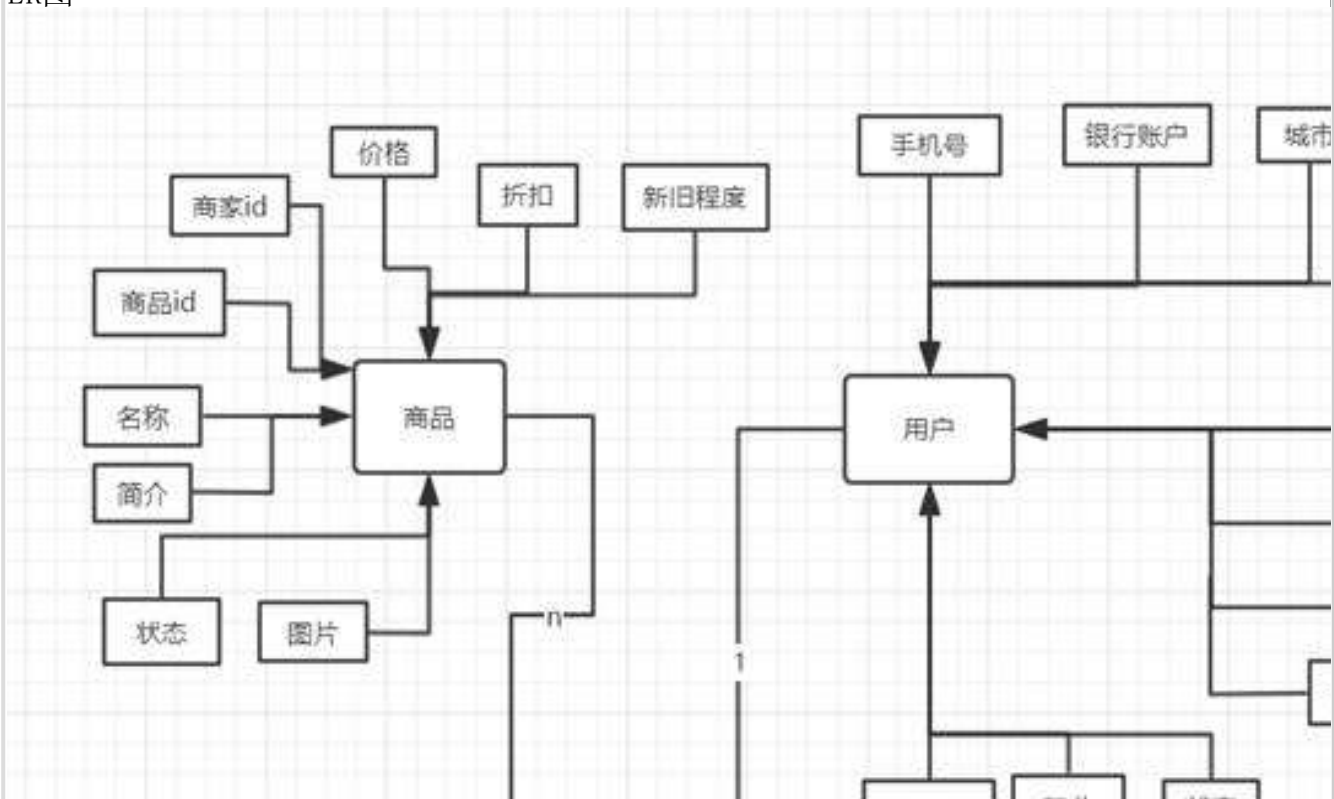
4.4 自由表述：

系统主要是提供在线商品的浏览和搜索，并提供用户下单购买的主要流程；同时提供用户注册个人信息和购买历史记录等信息。用户必须先申请成为商家才可以上架商品，商家必须先申请商品，一切审核由管理员完成，用户在购物车里下单商品，商家可以通过订单模块查看订单和评价。

5 系统数据结构设计

5.1 逻辑结构设计要点：

ER图



#	名称	数据类型
 1	id	INT
2	password	CHAR
3	username	VARCHAR
4	region	CHAR
5	loc	VARCHAR
6	tel	CHAR
7	isbusiness	TINYINT
8	score	TINYINT

2. 商品信息视图设计
用途：输入商品编号
输出对应商品信息

#	名称	数据类型
 1	goodsid	INT
2	name	VARCHAR
 3	id	INT
4	price	DOUBLE
5	num	INT
6	des	VARCHAR




外键

键名	字段	关联表	外联字段	UI
 FK_goods_student	id	secondhand.student	id	R

3. 商品评论信息视图设计
用途：输入商品编号
输出商品相关评论列表

外键

4. 购物车信息视图设计
用途：输入用户id
输出购物车内商品信息




键名	字段	关联表	外联字段	UP
 FK_cart_goods	goodsid	secondhand.goods	goodsid	R
 FK_cart_student	buyerid	secondhand.student	id	R
 FK_cart_student_2	sellerid	secondhand.student	id	R

5. 买家订单信息视图设计

用途：输入买家（卖家）信息
输出买家（卖家）对应的订单

#	名称	数据类型
 1	businessid	INT
 2	sellerid	INT
 3	buyerid	INT
 4	goodsid	INT
5	date	DATETIME
6	count	INT
7	iscomment	TINYINT

外键

键名	字段	关联表	外联字段	UP
 FK_business_goods	goodsid	secondhand.goods	goodsid	R
 FK_business_student	sellerid	secondhand.student	id	R
 FK_business_student_2	buyerid	secondhand.student	id	R

6. 卖家申请信息视图设计

用途：记录卖家申请

#	名称	数据类型
 1	id	INT
 2	stuid	INT
3	banknumber	VARCHAR

联合主键中的红色钥匙代表了连接了外表的

7. 管理员账号视图设计

用途：记录管理员账号

5.2 物理结构设计要点：

数据存储于Mysql数据库中，版本为14.14

1. 学生表【student】

loc	varchar (50)		详细地址
score	TINYINT (4)	非空	商店评分
isbusiness	TINYINT (1)	非空	是否为商家 【0: 不是商家】 【1: 是商家】

2. 商品表【goods】

列名	类型	备注	作用
goodsid	int	自增一，非空，主键	商品编号
id	int	非空，外键	商家学号
name	varchar (50)	非空	商品名
price	double	非空	价格
num	int	非空	数量
des	varchar (100)	非空	商品描述

3. 购物车表【cart】

列名	类型	备注	作用
buyerid	int	非空，联合主键	买家学号
goodsid	int	非空，联合主键	商品编号
sellerid	int	非空，外键	商家学号
num	int	非空	购物车中

4. 账单表【business】

列名	类型	备注	作用
businessid	int	自增一，非空，主键	账单编号
buyerid	int	非空，外键	买家学号
sellerid	int	非空，外键	商家学号
goodsid	int	非空，外键	商品编号
date	datetime	非空	购买时间
count	int	非空	购买数量
iscomment	tinyint (4)	非空	是否评价 【0: 已经评价】 【1: 还未评价】

5. 评论表【comments】

列名	类型	备注	作用
commentid	int	自增一，非空，主键	评论编号
content	varchar (100)	非空	评论内容
id	int	非空，外键	评论用户
time	datetime	非空	评论时间
goodsid	int	非空，外键	商品编号
sellerid	int	非空，外键	卖家编号
score	int		评分

6. 申请表【aply】

列名	类型	备注	作用
id	int	自增一，非空，主键	申请编号
stuid	int	非空，外键	申请同学学号
banknumber	varchar (50)	非空	银行卡号

7. 管理员表【admin】

列名	类型	备注	作用
id	int	非空，主键	管理员编号
password	varchar (50)	非空	密码

6 系统出错处理设计

6.1 错误处理

出错信息：

- 1、避免因用户操作不当和疏忽对系统造成的损坏；
- 2、采用对话框形式为用户提供警告信息；
- 3、一些关键性操作（比如：删除操作），都应该提供确认机制

补救措施：

- 故障出错后可能采取的交通措施，包括：
- a. 后备技术说明：采用廉价磁盘冗余阵列（RAID）技术，一旦原始系统数据丢失便启动副数据；
 - b. 降效技术说明：管理员在删除用户、警告用户采用的是人为浏览信息；
 - c. 恢复及再启动技术说明：当系统出现故障，需要重新启动计算机，并调用日志恢复文件，重新执行。

6.2 系统维护设计：

网站必须按照面向对象的软件工程的步骤进行开发，并充分考虑软件可维护性，软件的模块化、低耦合的特点，同时程序的内部必须有详细的注释和统一的编码格式、按时认真的进行代码的编写和备份工作，以利于未来版本，利于未来版本升级及移植等二次开发