库存管理系统

1.引言

1.1编写背景

传统的库存管理系统采用人工录入、管理方式，手续繁琐，容易出错，对操作人员的心智负担很重。此外，由于传统的系统采用纸质记录档案，管理不当极易损坏，且不方便不同人员异地协同管理，极大的限制了企业库存系统的健壮性和可用性。

1.2编写目的

为了提高效率、推进自动化管理水平，我们需要编写一套基于B/S构建的自动化管理的库存管理系统。

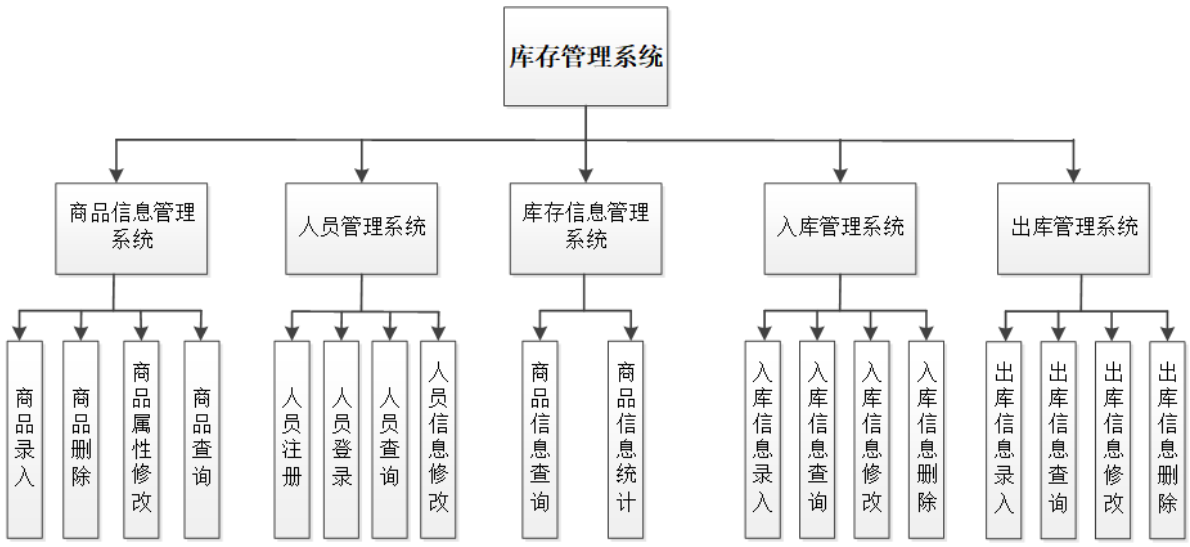
2.系统任务概述

2.1目标

本系统可以细化为五个子系统：

- 商品信息管理系统
- 人员管理系统
- 入库管理系统
- 出库管理系统
- 库存信息管理系统

层次结构图：



2.2 用户特点

- 划分权限为 管理员(Admin) 和 普通用户 (User)
- 有着传统的库存管理经验
- 缺乏信息化库存管理经验

3.需求规定

3.1 对功能的规定

- 管理员：
    1. 增删查改商品信息
    2. 修改所有用户信息
    3. 增删查改出入库信息
    4. 查询统计商品信息
- 用户：
    1. 增删查改出入库信息
    2. 查询统计商品信息

3.2对性能的规定

- Server端响应时间小于`500ms`（包括与数据库的交互）
- Browser端首屏加载时间小于`1s`
- 网络时延小于`3s`

4.运行环境规定

- 服务器：Ubuntu Server 20.04.1 LTS
- 运行时：
    - .NET Core 3.1
    - Node.js 12.18.4 LTS
- 数据库：MySQL 8
- 浏览器：

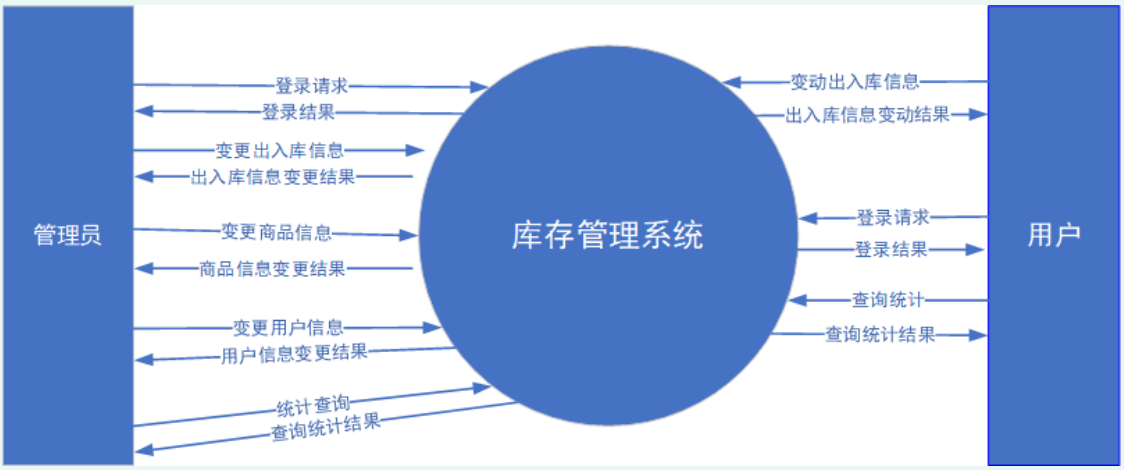
浏览器	版本
IE	10+
Edge	12+
Firefox	21+
Chrome	23+
Safari	6.1+
Android	4.4+

5.数据流程、功能描述和设计

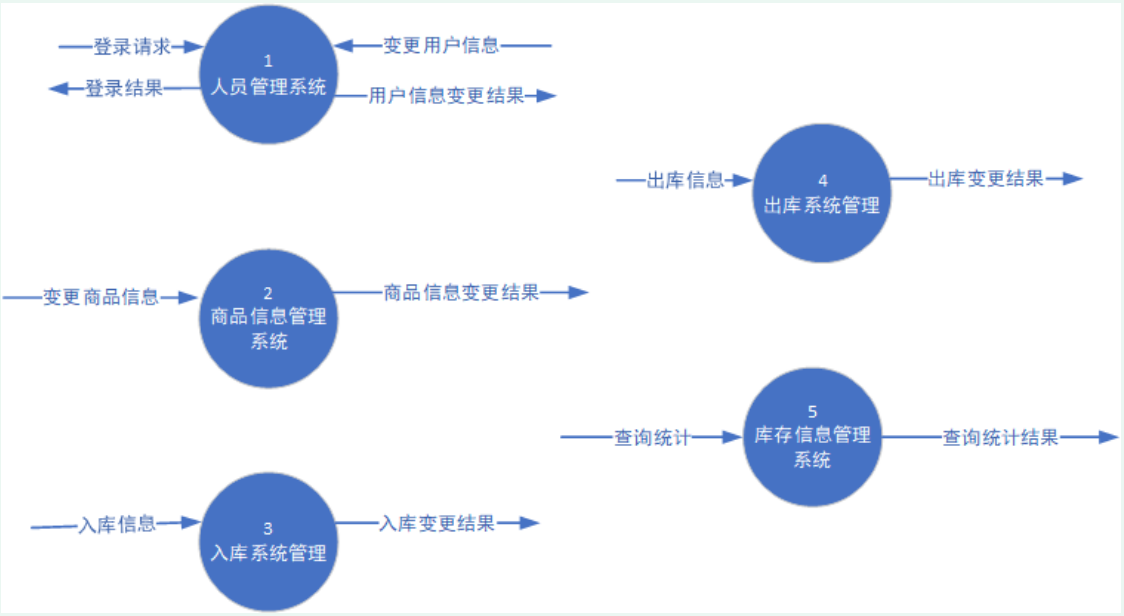
> 给出数据流图并分解
>
> 提供数据字典
>
> 数据流描述

5.1数据流图

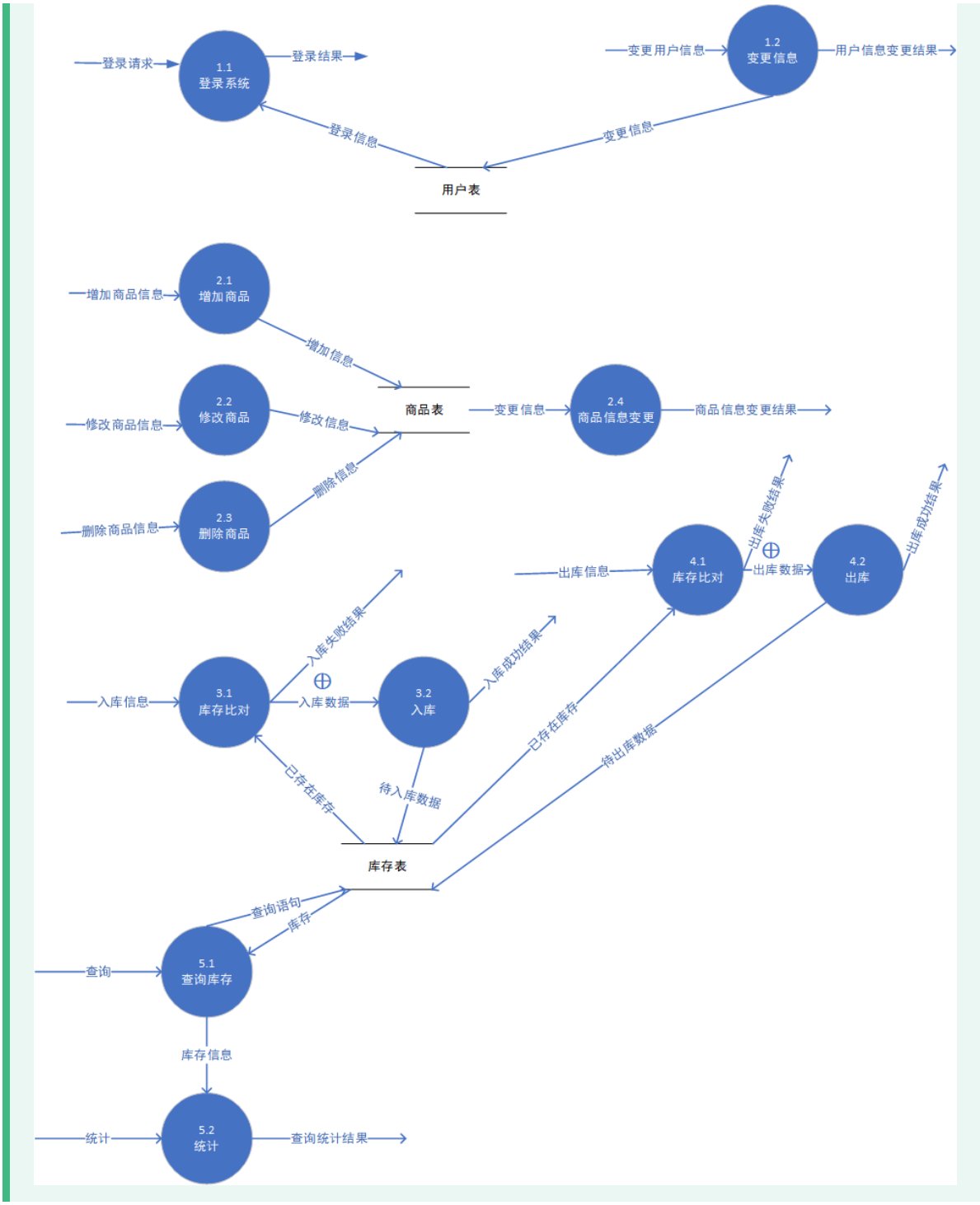
顶层：



0层：



1层：



5.2 数据描述

5.2.1 数据库描述

表5.2-1 用户表

字段名	描述	类型	长度	是否可以为空	是否为主键
usr	用户名	nvarchar	20	no	yes
pwd	密码	varchar	20	no	no
authority	权限	int	-	no	no

表5.2-2 商品表

字段名	描述	类型	长度	是否可以为空	是否为主键
no	商品编号	int	-	no	yes
name	商品名称	nvarchar	100	no	no
price	价格	float	-	no	no
type	类别	nvarchar	10	no	no
date	生产日期	datetime	-	no	no
lastdate	保质期(天)	int	-	no	no
company	所在公司	nvarchar	100	no	no
num	库存数量	int	-	no	no

表5.2-3 库存表

字段名	描述	类型	长度	是否可以为空	是否为主键
ddno	订单号	int	-	no	yes
no	商品编号	int	-	no	no(外键)
usr	操作者用户名	nvarchar	20	no	no(外键)
opNum	操作数量 （正数为入库，负数为出库）	int	-	no	no(check)

5.2.2 数据字典

由**数据流图**和**层次结构图**中设计的数据定义数据字典如下：

1. 登录名 = 1{字符}20
2. 密码 = 1{字母|数字}20
3. token = 1{字母|数字}
4. 商品编号 = 1{数字}20
5. 商品名称 = 1{字符}100
6. 价格 = 数字 + {.} + 数字
7. 类别 = 1{字符}10
8. 生产日期 = 日期
9. 保质期 = 数字
10. 所在公司 = 1{字符}100
11. 库存数量 = 非负整数
12. 订单号 = 数字
13. 操作数量 = 数字
14. 登录请求 = 登录名 + 密码
15. 登陆结果 = token + 登录信息
16. 登录信息 = 0{字符}50
17. 变更用户信息 = 登录名 + 密码 + token
18. 用户信息变更结果 = 0{字符}50
19. 登录信息 = 登录名 + 密码
20. 变更信息 = 登录名 + 密码

21. 增加商品信息 = token + 商品编号 + 商品名称 + 价格 + 类别 + 生产日期 + 保质期 + 所在公司 + 库存数量
22. 修改商品信息 = token + 商品编号 + 商品名称 + 价格 + 类别 + 生产日期 + 保质期 + 所在公司 + 库存数量
23. 删除商品信息 = token + 商品编号
24. 增加信息 = 商品编号 + 商品名称 + 价格 + 类别 + 生产日期 + 保质期 + 所在公司 + 库存数量
25. 修改信息 = 商品编号 + 商品名称 + 价格 + 类别 + 生产日期 + 保质期 + 所在公司 + 库存数量
26. 删除信息 = 商品编号
27. 商品信息变更结果 = 0 { 字符 } 50
28. 入库信息 = token + 订单号 + 商品编号 + 操作数量
29. 入库数据 = 订单号 + 登录名 + 商品编号 + 操作数量
30. 入库失败结果 = 0 { 字符 } 50
31. 入库成功结果 = 0 { 字符 } 50
32. 待入库数据 = 订单号 + 商品编号 + 登录名 + 操作数量
33. 已存在库存 = 商品编号 + 操作数量
34. 出库信息 = 入库信息
35. 出库失败结果 = 入库失败结果
36. 出库数据 = 入库数据
37. 出库成功结果 = 入库成功结果
38. 待出库数据 = 待入库数据
39. 查询 = token + 订单号 + 价格 + 类别 + 所属公司 + 字符
40. 查询语句 = 订单号 + 价格 + 类别 + 所属公司 + 字符
41. 库存信息 = 入库信息 + 出库信息
42. 库存 = 库存信息
43. 统计 = 0 { 字符 } 50
44. 查询统计结果 = 库存信息 + 登录名