**云南大学滇池学院理工学院**

**2022- 2023学年第一学期**

**课程论文**

**题 目 药品管理系统**

**课程名称 人工智能综合实践**

**年级专业 人工智能**

**学 号 20202161008**

**姓 名 陈思韬**

**教师评语：**

**成绩评定： 分 任课教师签名：**

### 课程设计（大作业）任务书

姓 名：陈思韬 院（系）：理工学院

专 业：人工智能 学 号：20202161008

任务起止日期：2023/6/2-2023/6/25

课程设计题目： 药品管理系统

课程设计要求：

1. 系统功能定义：明确药品管理系统的功能， ①提供了不同批次药品的唯一编号，通过python中的uuid来实现，确保了同一种药物可能出现的生产日期不同的问题，为处理临期药物，过期药物提供了方便 ②药品信息录入（通过手动输入药品名字，价格，编号，生产日期，药品种类，货架，实际生活应该可以直接通过条形码识别来导入信息，这里由于技术限制，暂且采用一般化的策略）， ③库存管理（应当记录药品在仓库中的数量，在入库，出库时，对药品数量进行相应的操作。。注意：在记录时数量的变更应该是针对入库，出库时候，实际的药品数量，而不能是根据次数，因为每一次药品的数量都可能不同） ④药品生产日期以及到期时间的记录与管理（应当对于过期或者临期药品进行处理，所以需要记录生产日期和到期时间） ⑤药品信息查询功能（输入药品名字进行查询，展示药品的剩余数量，以及在仓库中的所在位置）， ⑥药品数量不够时需要补齐的提醒(当仓库中的药品数量<=50的时候提醒仓库需要增加药品)

⑦药品分类与标识，在输入药品时，可以输入药品种类，（一个药品可以有多个种类）

1. 数据结构设计：设计合适的数据结构来存储药品信息和相关数据。使用类，字典，列表等数据结构来组织数据，如药品信息可以用字典（药品名字，价格，编号）
2. 用户界面设计：通过合适的界面来使得用户可以进行相关的操作，（具体界面使用终端那种文字的，还是图形化的有待确定）
3. 功能实现：根据系统功能定义，使用python来实现各个功能模块，包括药品信息的添加，更新，查询，库存管理，销售记录的记录和查询，药品数量的查询，药品所在仓库中位置的查询。
4. 异常：防止用户输入错误，如输入负数的药品数量，负数的库存数量
5. 文件存储：使用文件读写操作，将操作药品信息和相关数据保存到文件中，在每次运行药品管理系统时读取文件，导入之前的数据，使得管理系统具有记忆性，可以持续化的去运行，（当然也应该提供一个初始化的模块。来初始化药品管理系统中的相关信息）

工作计划及安排：

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 完成内容 | 时间 | 备注 |
| 1 | 布置任务、讲授设计 | 6/2 |  |
| 2 | 总体设计 | 6/8 |  |
| 3 | 详细设计 | 6/12 |  |
| 4 | 测试 | 6/16 |  |
| 5 | 书写设计报告 | 6/20 |  |
| 6 | 验收、考核 | 6/25 |  |
| 合计 | | 23天 |  |

指导教师签字

年 月 日

**课程设计（大作业）成绩**

学号：20202161008 姓名：陈思韬 指导教师：冯微玮

课程设计题目： 基于python的药品管理系统

总结：主要通过python设计了一个药品管理系统，系统主要分为用户登录，输入药品信息，药品的库存管理，药品的过期查询，药品数量的查询，药品的信息查询，查询种类

在实验中遇到了一些问题，比如：①输入药品信息时，有对比系统时间的需求，但让用户输入要求的格式不理想，所以通过分开输入，再合并，实现格式的要求，②是批次号，最初我采用的是uuid模块随机生成批次号，但后来老师指出，这种是不方便的，我在网上查询相关信息后，采用了名字\_年份\_日期作为批次号，（标准的有国药准字什么的，这里实现起来比较复杂，因此采用了更易懂同时也满足需求的实现）③在实现库存管理时，由于程序设计的理念是先输入药品编号，再想办法找药品批次号，经历了几个阶段（从最开始的需要到txt文件中去复制粘贴，再到后来的在输出中展示出来再复制，到现在的自动给有相同药品编号的批次号前面加上一个序号，使得用户可以直接选择进行输入）④在最后实现展示文件内容的时候，原本的文件内容展示出来过于拥挤，因此采用计算每个字段的最大长度，然后使用 “ | ”符号把内容分割开来，使得内容展示更加的易读。

指导教师评语：

成绩：

填表时间：2023年6月28日 指导教师签名：

《人工智能综合实践》课程设计报告

**一、设计要求**

**1、题目分析**

根据要求，设计一个药品管理系统。该系统旨在提供一个便捷、高效的药品信息管理平台，以满足用户对药品存储、药品信息查询，库存管理等需求。

该药品管理系统的主要目标是实现以下功能：

药品信息管理：系统应能够记录、存储和管理药品的相关信息，包括药品名称、药品编号、生产厂商、单价、有效期等。用户可以通过系统添加、删除、修改和查询药品信息。

销售管理：系统应具备销售记录的功能，能够记录每一次销售的日期、销售数量、销售金额等信息。销售功能的同时，系统应自动更新相关药品的库存数量，并及时发出库存不足的警告。

库存管理：系统应提供库存管理功能，即实时跟踪和更新药品的库存数量。用户可以通过系统查询当前药品的库存情况，并且系统应能够根据销售记录自动更新库存数量。

1. **需求分析**
   1. 药品信息录入，要有药品的名字，药品编号，药品的生产日期，药品的有效日期，药品数量，药品种类（一个药品可能有多种标签，如布洛芬（退烧药，止痛药）
   2. 用户登陆，设计了两个用户，分别是root和user，对于root用户，有六个功能，药品信息录入，库存管理，药品生产日期和其他管理，药品信息查询功能，药品数量不够需要补充的提醒，药品分类与表示，而对于user用户，有药品信息录入，药品生产日期和其他管理，药品信息查询功能，药品分类与标识功能，共四个
   3. 对于同种药品的信息录入，由于同种药品可能存在生产日期不同的问题，因此录入的时候，利用python的uuid，为每个批次的药品分配唯一的编号
   4. 药品生产日期和其他管理，对于药品管理系统，临期药品和过期药品需要被处理，因此，查询这些药品的信息就成了必要，通过药品的过期时间与系统时间的对比，从而得出临期药品和过期药品
   5. 药品信息查询功能，提供了药品信息的查询，通过药品编号来实现（不是uuid赋给药品的编号）
   6. 药品数量不够需要补充，当药品数量小于等于50的时候，会生成一份名为需要补充药品的txt文件，里面有相关信息。
   7. 药品分类标识，药品在入库的时候会写入一个或多个种类标签，查询的时候直接输入标签查询，来实现药品的分类查询

**二、概要设计**

1、主界面设计

通过python代码简单的时间一个管理系统的初始界面，不同用户对应着不同的菜单界面

存储结构设计

1. 用户账户存储结构设计：

-使用一个字典来存储用户账户信息，其中键（key）是用户名，值（value）是对应的密码。例如，`accounts = {'root': '123456', 'user': '789456'}`。这样可以在登录过程中验证用户的用户名和密码是否正确。

2. 药品信息存储结构设计：

可以将药品信息存储在一个文本文件（例如'药品清单.txt'）中，每行表示一个药品的信息。

每行的药品信息可以使用逗号分隔的方式进行存储，包括以下字段：

药品名称（drug\_name）

药品编号（drug\_number）

药品生产日期（drug\_production\_date）

药品有效期（drug\_expiry\_date）

批次号（batch\_number）

数量（quantity）

货架号(drug\_shelf)

层数(drug\_level)

药品种类（drug\_categories）

在需要更新或查询药品信息时，可以读取文件并将每行解析为对应的药品信息。

3. 批次号生成和存储：

在药品信息录入过程中，可以使用`generate\_batch\_number()`函数生成唯一的批次号，该批次号可以包含药品名称的一部分和唯一标识符。

批次号可以作为药品信息的一个字段存储在药品信息中，用于标识不同批次的药品。

4. 库存管理：

库存管理涉及药品的入库和出库操作，可以在药品信息中添加一个字段来记录药品的库存数量。

当进行入库或出库操作时，可以更新对应药品的库存数量，并将更新后的药品信息存储回文本文件中。

5. 临期药品和过期药品管理：

可以在药品信息中添加字段来记录药品的生产日期和有效期。

在药品生产日期和其他管理菜单中，可以根据当前日期和药品的生产日期和有效期，判断哪些药品处于临期或过期状态，并进行相应的提醒或显示。

6.药品分类与标识药品分类标识，药品在入库的时候会写入一个或多个种类标签，查询的时候直接输入标签对应的序号查询，来实现药品的分类查询

**3、系统功能设计**

（1多用户登陆

（2）药品唯一批次号

（3）药品生产日期管理

（4）库存管理

（5）药品生产日期和其他管理

（6）药品分类管理

（7）药品信息录入

（8）药品信息查询功能

（9）药品数量不够需要补充提醒

……

**三、模块及其功能设计**

**1. 模块设计：**

**(1) 模块1：登录模块**

**- 函数：login()**

**- 功能：实现用户登录功能，根据输入的用户名和密码验证登录信息，并根据用户类型显示相应的菜单选项。**

**(2) 模块2：药品信息录入模块**

**- 函数：drug\_information\_entry()**

**- 功能：从文件中读取已有药品信息，接收用户输入新的药品信息并添加到列表中，最后将更新后的药品信息写回文件。**

**(3) 模块3：库存管理模块**

**- 函数：inventory\_management()**

**- 功能：根据用户选择，实现药品的入库和出库操作，更新药品的数量信息，并将更新后的药品信息写回文件。**

**(4) 模块4：药品生产日期和其他管理模块**

**- 函数：drug\_date\_management()**

**- 功能：根据用户选择，显示临期和过期药品的信息，包括药品编号、名称、生产日期、有效期等。**

**(5) 模块5：药品信息查询模块**

**- 函数：drug\_information\_query()**

**- 功能：根据用户输入的药品编号，查询并显示对应药品的详细信息，包括药品名称、编号、生产日期、有效期、批次号和数量等。**

**(6) 模块6：药品分类与标识模块**

**- 函数：drug\_category\_query()**

**- 功能：根据用户输入的药品种类序号，查询并显示该类药品的信息，包括药品名称、编号、生产日期、有效期、批次号和数量等。**

**2. 系统子程序级功能设计：**

1. **子程序一：登陆，通过字典实现对于不同用户的登陆**
2. **子程序二：generate\_batch\_number(),uuid，通过uuid来生成对于同名不同批次药品的唯一编号**
3. **子程序三：show\_root\_menu 共有七个功能，用户输入不同的数字，实现不同的功能。**
4. **子程序四：show\_user\_menu 共有五个功能，相比于root用户少了两个功能**
5. **子程序五：drug\_informataion\_entry() 用于实现药品信息输入，输入药品名称，编号，生产日期，有效期，药品数量，药品种类**
6. **子程序六：inventory\_management()，库存管理，实现出库，入库功能，调用函数updata\_inventory(drug\_number,drug\_uuid,quantity,’operation’)来实现**
7. **子程序七：updata\_inventory(drug\_number,drug\_uuid,quantity,’operation’)，实现了药品清单的更新，**
8. **子程序八：drug\_date\_management()日期管理，读取文件，将药品信息转化为列表，通过计算药品的过期时间和当前系统时间的差来判断是过期药品还是临期药品。**
9. **子程序九：drug\_information\_query() 打开文件，遍历，将信息存储到drugs列表中，便利drugs，找到和输入的drug\_number相等的药品编号**
10. **子程序十：check\_low\_quantity\_drugs()检查药品质量，把所有药品数量不足50的药品的药品信息写入到需要补充药品.txt文件中**
11. **子程序十一：drug\_category\_query() 输入药品种类，得出对应种类下所有药品信息。**
12. **子程序十二：show\_file\_content()展示药品信息，通过竖线分割开，使得展示结果更加美观**
13. **子程序十三: show\_file（），选择要展示哪一个文件的信息。调用子程序十二**
14. **模块1功能：实现用户登录**

**-函数：login()**

**-功能：通过字典存储的用户信息，进行键值对对应，实现密码登陆。**

**(2) 模块2功能：药品信息录入功能**

**- 函数：drug\_information\_entry()**

**- 功能：接收用户输入的药品信息，包括药品名称、编号、生产日期、有效期、数量和药品种类，将信息添加到药品列表中，并将更新后的药品信息写回文件。**

**(3) 模块3功能：库存管理功能**

**- 函数：inventory\_management()**

**- 功能：根据用户选择，实现药品的入库和出库操作，更新药品的数量信息，并将更新后的药品信息写回文件。**

**(4) 模块4功能：药品生产日期和其他管理功能**

**- 函数：drug\_date\_management()**

**- 功能：根据用户选择，显示临期和过期药品的信息，提醒药品的管理情况。**

**(5) 模块5功能：药品信息查询功能**

**- 函数：drug\_information\_query()**

**功能：根据用户输入的药品编号，查询并显示对应药品的详细信息，包括药品名称、编号、生产日期、有效期、批次号和数量等。**

**(6) 模块6功能：药品分类与标识功能**

**- 函数：drug\_category\_query()**

**- 功能：根据用户输入的药品种类，查询并显示该类药品的信息，包括药品名称、编号、生产日期、有效期、批次号和数量,货架号，层数等。**

**(7)模块7功能：药品清单展示**

**-功能：用户选择要输出的文件，进行内容展示**

**3.函数调用关系：**

Login()

输入用户名和密码

输入正确，登陆成功！

登陆的是root用户

登陆的是user用户

用户或密码错误，请重新输入

Login()

show\_root\_menu()

Show\_user\_menu()

drug\_information\_entry()

inventory\_management()

check\_low\_quantity\_drugs()

show\_file()

drug\_category\_query()

drug\_information\_query()

drug\_date\_management()

1

2

3

4

return

5

6

7

8

1

2

3

4

5

药品信息录入

打开文件，使用line.strip()规范格式

文件不存在，FileNotFoundError

开始输入药品的相关信息

将输入的信息保存到药品信息.txtx文件中

输入不规范

库存管理

读取成功

读取失败

入库操作

出库操作

输入药品编号和药品批次号

输入入库数量

输入药品编号和选择药品批次号

输入出库数量，

update\_inventory(drug\_number, drug\_uuid, quantity\_change, operation)

药品日期管理

读取成功

读取失败

显示临期药品

显示过期药品

返回

输入年月日与系统时间做比较

大于0小于45

小于0

输出临期药品信息

输出过期药品信息

药品信息查询

输入药品编号

遍历存储列表，输出所有编号为输入编号的药品信息

根据种类查询药品信息

输出数量不足药品信息

便利列表，寻找数量小于等于50的药品

把得到的药品信息存入需要补充药品.txt文件中

输入药品种类，遍历列表，输出有这个标签的所有药品的信息

展示文档信息

展示药品信息.txt

展示需要补充药品.txt

**四、详细设计**

**1、数据类型定义**

**根据你提供的代码，我们可以看到以下数据类型的使用：**

**str（字符串）：在代码中用于接收用户输入的用户名和密码，例如`username`和`password`。**

**dict（字典）：`accounts`变量是一个字典，用于存储用户名和密码的键值对。**

**int（整数）：在代码中用于接收药品数量的输入，例如`drug\_quantity`和`quantity`。**

**list（列表）：`drug\_categories`是一个列表，用于存储药品的种类。**

**datetime（日期时间）：代码中使用`datetime`模块来处理日期和时间相关操作，例如计算药品的剩余天数。**

**uuid（唯一标识符）：使用`uuid`模块生成唯一的批次号。**

**2、系统主要子程序详细设计**

**（1）主程序模块设计**

**主函数**

def login():  
 accounts = {'root': '123456', 'user': '789456'}  
  
 while True:  
 username = input('请输入用户名：')  
 password = input('请输入密码：')  
  
 if username in accounts and accounts[username] == password:  
 print('登录成功！')  
 if username == 'root':  
 show\_root\_menu()  
 else:  
 show\_user\_menu()  
 else:  
 print('用户名或密码错误，请重新输入。')

**（2）子程序一**

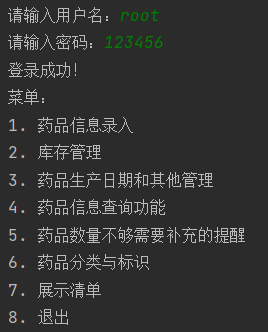
**（3）子程序二**

**……**

**五、测试分析**

**下面给出每个子程序的调试截图**

**（1）主程序模块设计**

**截图一**

**分析：药品管理系统设计了如上截图所示的功能，实现了多用户登陆，药品信息的录入，药品的库存管理，过期，临期药品的查询，药品信息查询功能，药品数量不够需要补充的提醒，药品种类的查询。**

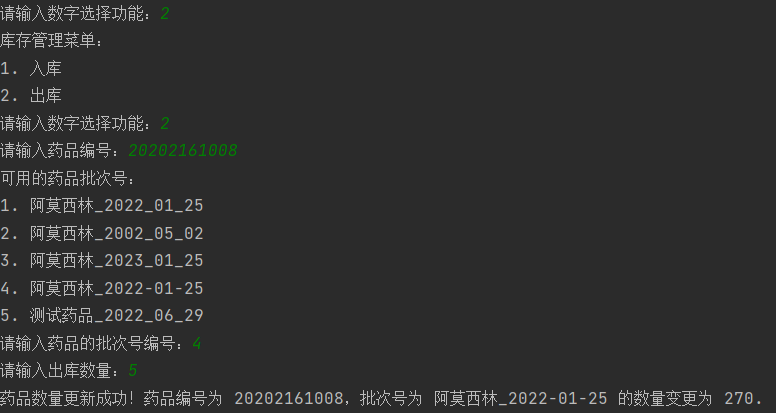
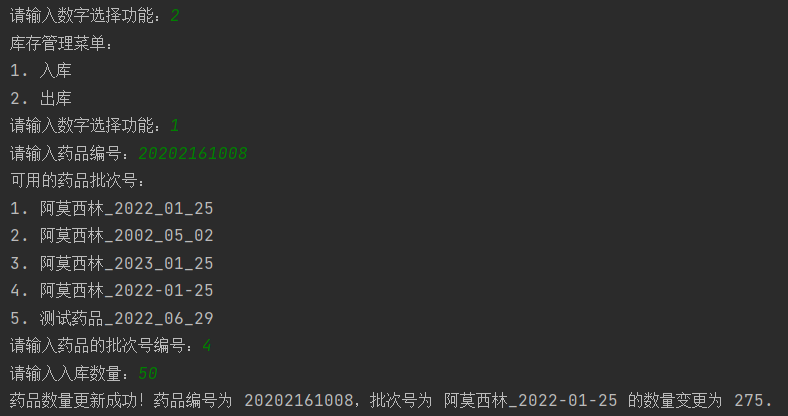
**（2）子程序一**

**截图二**



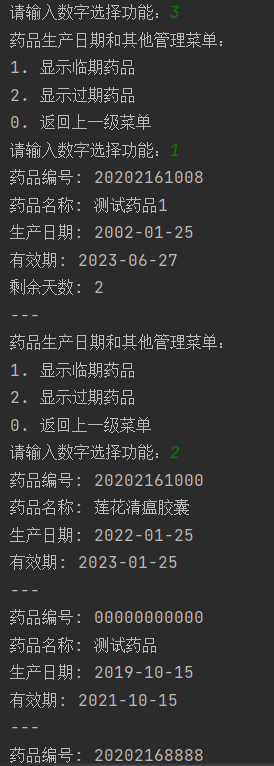
**分析：药品信息的录入功能，输入药品的多个信息，为查询药品时提供了关键索引**

**（3）子程序二**

**截图三**

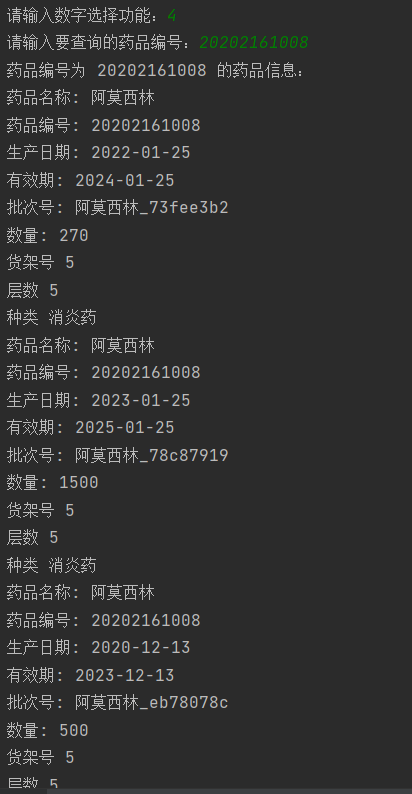
**分析：库存管理分为两个子模块，分别是入库和出库，通过调用相关的函数来实现药品的出入库管理**

**子程序四：**

**截图四：**

**分析：药品生产日期和其他管理，主要有两个功能，展示临期药品和过期药品，临期药品的判断条件是药品过期时间距离系统时间大于0小于45，而过期药品则是小于0**

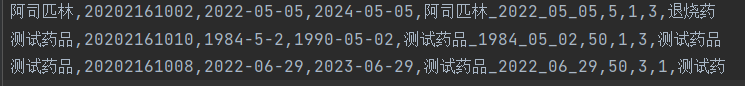
**子程序五：**

**截图五：**

**分析：查询药品，通过输入药品编号，会输出相同编号药品的所有不同批次**

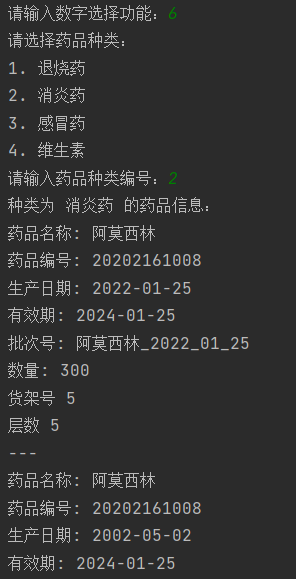
子程序六：

截图六：



**分析：选择需要补充药品信息功能后，系统会自动生成一份txt文件，里面包含了需要补充的药品的信息**

**子程序七：**

**截图七：**

**分析：通过输入药品时的类别关键字，来进行药品的查询，同一类别下可以展示出多种药品**

**六、源程序清单**

**（下面的程序用小五号字排版）**

**（1）程序初始化**

**（2）子程序一**

def generate\_batch\_number(drug\_name, drug\_production\_year, drug\_production\_month, drug\_production\_day):  
 batch\_number = f"{drug\_name}\_{drug\_production\_year}\_{drug\_production\_month:02}\_{drug\_production\_day:02}"  
  
 return batch\_number

**(3)子程序二**

def login():  
 accounts = {'root': '123456', 'user': '789456'}  
  
 while True:  
 username = input('请输入用户名：')  
 password = input('请输入密码：')  
  
 if username in accounts and accounts[username] == password:  
 print('登录成功！')  
 if username == 'root':  
 show\_root\_menu()  
 else:  
 show\_user\_menu()  
 else:  
 print('用户名或密码错误，请重新输入。')

if username == 'root':  
 show\_root\_menu()  
 else:  
 show\_user\_menu()  
 else:  
 print('用户名或密码错误，请重新输入。')

1. **子程序三**

def show\_root\_menu():  
 while True:  
 print('菜单：')  
 print('1. 药品信息录入')  
 print('2. 库存管理')  
 print('3. 药品生产日期和其他管理')  
 print('4. 药品信息查询功能')  
 print('5. 药品数量不够需要补充的提醒')  
 print('6. 药品分类与标识')  
 print('7. 展示清单')  
 print('8. 退出')  
  
 choice = input('请输入数字选择功能：')  
 if choice == '1':  
 drug\_information\_entry()  
 elif choice == '2':  
 inventory\_management()  
 elif choice == '3':  
 drug\_date\_management()  
 elif choice == '4':  
 drug\_information\_query()  
 elif choice == '5':  
 check\_low\_quantity\_drugs()  
 elif choice == '6':  
 drug\_category\_query()  
 elif choice == '7':  
 show\_file()  
 elif choice == '8':  
 print('退出')  
 return  
 else:  
 print('无效的选择，请重新输入。')

1. **子程序四**

def show\_user\_menu():  
 print('菜单：')  
 print('1. 药品信息录入')  
 print('2. 药品生产日期和其他管理')  
 print('3. 药品信息查询功能')  
 print('4. 药品分类与标识')  
 print('5.退出')  
 # 在这里添加其他功能选项  
 choice = input('请输入数字选择功能：')  
 if choice == '1':  
 drug\_information\_entry()  
 elif choice == '2':  
 drug\_date\_management()  
 elif choice == '3':  
 drug\_information\_query()  
 elif choice == '4':  
 drug\_category\_query()  
 elif choice == '5':  
 print('退出')  
 return  
 else:  
 print('无效的选择，请重新输入')

**(6)子程序五**

def drug\_information\_entry():#药品信息录入  
 file\_path = '药品清单.txt'  
 drugs = []  
  
 try:  
 with open(file\_path, 'r') as file:  
 for line in file:  
 drug\_info = line.strip().split(',')  
 # 使用line.strip()方法去除首尾的空白字符，然后使用split(',')方法将该行内容按逗号进行分割，生成一个药品信息的列表。  
 drugs.append(drug\_info)  
 except FileNotFoundError:  
 print('药品信息文件不存在。')  
 return  
  
 drug\_name = input('请输入药品名称：')  
  
 while True:  
 try:  
 drug\_number = input('请输入药品编号：')  
 if len(drug\_number) != 11:  
 raise ValueError('药品编号长度应为11位')  
 if not drug\_number.isdigit():  
 raise ValueError('药品编号应为纯数字')  
 break # 输入正确，退出循环  
 except ValueError as e:  
 print(str(e))  
  
 # 此时药品编号长度等于11且只包含数字，将输入的药品编号赋给变量 drug\_number  
 print('药品编号正确。')  
  
  
 drug\_production\_year = input('请输入药品生产日期的年：')  
 drug\_production\_month = input('请输入药品生产日期的月：')  
 drug\_production\_day = input('请输入药品生产日期的天：')  
  
 drug\_expiry\_year = input('请输入药品有效期的年份')  
 drug\_expiry\_month = input('请输入药品有效期的月份')  
 drug\_expiry\_day = input('请输入药品有效期的天')  
  
 while True:  
 drug\_shelf = int(input('请输入货架号：'))  
 if drug\_shelf < 1 or drug\_shelf > 10:  
 print('货架号超出范围，请重新输入。')  
 else:  
 break  
  
 while True:  
 drug\_level = int(input('请输入要存放的层数：'))  
 if drug\_level < 1 or drug\_level > 10:  
 print('层数超出范围，请重新输入。')  
 else:  
 break  
 drug\_shelf = str(drug\_shelf)  
 drug\_level = str(drug\_level)  
 # 将年、月、日合并成日期字符串  
 drug\_production\_date = f"{drug\_production\_year}-{drug\_production\_month}-{drug\_production\_day}"  
 drug\_expiry\_date = f"{drug\_expiry\_year}-{drug\_expiry\_month}-{drug\_expiry\_day}"  
  
  
 while True:  
 try:  
 drug\_quantity = input('请输入药品数量：')  
 if not drug\_quantity.isdigit():  
 raise ValueError('药片数量应该为数字')  
 break  
 except ValueError as e:  
 print(str(e))  
  
 drug\_categories = []  
  
 while True:  
 category = input('请输入药品种类（输入0结束输入）：')  
 if category == '0':  
 break  
 drug\_categories.append(category)  
  
 batch\_number = generate\_batch\_number(drug\_name, drug\_production\_year, drug\_production\_month, drug\_production\_day)  
 drug\_info = [drug\_name, drug\_number, drug\_production\_date, drug\_expiry\_date, batch\_number, str(drug\_quantity),drug\_shelf,drug\_level]  
 drug\_info.extend(drug\_categories)  
#这里使用extend是为了防止drug\_info的结构变得更加复杂，  
 drugs.append(drug\_info)  
  
#也使用。 drug\_info = [drug\_name, drug\_number, drug\_production\_date, drug\_expiry\_date, batch\_number, str(drug\_quantity)]  
 # for category in drug\_categories:  
 # drug\_info.append(category)  
  
 with open(file\_path, 'w') as file:  
 for drug in drugs:  
 file.write(','.join(drug) + '\n')  
  
 print('药品信息录入成功！')  
  
 drugs.append(drug\_info)

**(7)子程序六**

def inventory\_management():  
 file\_path = '药品清单.txt'  
 drugs = [] # 空列表用于存储药品信息  
  
 try:  
 with open(file\_path, 'r') as file:  
 for line in file:  
 drug\_info = line.strip().split(',')  
 drugs.append(drug\_info)  
 except FileNotFoundError:  
 print('药品信息文件不存在。')  
 return  
  
 print('库存管理菜单：')  
 print('1. 入库')  
 print('2. 出库')  
 choice = input('请输入数字选择功能：')  
  
 if choice == '1':  
 while True:  
 try:  
 drug\_number = input('请输入药品编号：')  
 if len(drug\_number) != 11:  
 raise ValueError('药品编号长度应为11位')  
 if not drug\_number.isdigit():  
 raise ValueError('药品编号应为纯数字')  
 break # 输入正确，退出循环  
 except ValueError as e:  
 print(str(e))  
  
  
 available\_batches = []  
 for batch\_info in drugs:  
 if len(batch\_info) > 1 and batch\_info[1] == drug\_number:  
  
 available\_batches.append(batch\_info[4])  
  
 print('可用的药品批次号：')  
 for i, batch in enumerate(available\_batches):  
 print(f"{i+1}. {batch}")  
  
 while True:  
 try:  
 batch\_choice = int(input('请输入药品的批次号编号：'))  
 if 1 <= batch\_choice <= len(available\_batches):  
 break  
 else:  
 raise ValueError('无效的选项')  
 except ValueError as e:  
 print(str(e))  
  
 selected\_batch = available\_batches[batch\_choice - 1]  
  
 while True:  
 try:  
 quantity = input('请输入入库数量：')  
 if not quantity.isdigit():  
 raise ValueError('入库数量应为纯数字')  
 break # 输入正确，退出循环  
 except ValueError as e:  
 print(str(e))  
  
 update\_inventory(drug\_number, selected\_batch, quantity, '入库')  
  
 elif choice == '2':  
 while True:  
 try:  
 drug\_number = input('请输入药品编号：')  
 if len(drug\_number) != 11:  
 raise ValueError('药品编号长度应为11位')  
 if not drug\_number.isdigit():  
 raise ValueError('药品编号应为纯数字')  
 break # 输入正确，退出循环  
 except ValueError as e:  
 print(str(e))  
  
 available\_batches = []  
 for batch\_info in drugs:  
 if len(batch\_info) > 1 and batch\_info[1] == drug\_number:  
 available\_batches.append(batch\_info[4])  
  
 print('可用的药品批次号：')  
 for i, batch in enumerate(available\_batches):  
 print(f"{i + 1}. {batch}")  
  
 while True:  
 try:  
 batch\_choice = int(input('请输入药品的批次号编号：'))  
 if 1 <= batch\_choice <= len(available\_batches):  
 break  
 else:  
 raise ValueError('无效的选项')  
 except ValueError as e:  
 print(str(e))  
  
 selected\_batch = available\_batches[batch\_choice - 1]  
  
 while True:  
 try:  
 quantity = input('请输入出库数量：')  
 if not quantity.isdigit():  
 raise ValueError('出库数量应为纯数字')  
 break # 输入正确，退出循环  
 except ValueError as e:  
 print(str(e))  
  
 update\_inventory(drug\_number, selected\_batch, quantity, '出库')  
  
 else:  
 print('无效的选项。')

**(8)子程序七**

def update\_inventory(drug\_number, drug\_uuid, quantity\_change, operation):  
 file\_path = '药品清单.txt'  
 drugs = []  
 quantity\_change = int(quantity\_change)  
  
 try:  
 with open(file\_path, 'r') as file:  
 for line in file:  
 drug\_info = line.strip().split(',')  
 drugs.append(drug\_info)  
 except FileNotFoundError:  
 print('药品信息文件不存在。')  
 return  
  
 updated\_drugs = []  
 updated = False  
  
 for drug in drugs:  
 if len(drug) > 4 and drug[1] == drug\_number and drug[4] == drug\_uuid:  
  
 if len(drug) > 5:  
 current\_quantity = int(drug[5])  
 else:  
 current\_quantity = 0  
  
 if operation == '出库' and current\_quantity < quantity\_change:  
 print('药品数量不足，无法出库。')  
 print(f'出库数量超过了可用数量！药品编号为 {drug\_number}，批次号为 {drug\_uuid} 的可用数量为 {current\_quantity}.')  
 return  
  
 if operation == '出库':  
 new\_quantity = current\_quantity - quantity\_change  
 elif operation == '入库':  
 new\_quantity = current\_quantity + quantity\_change  
  
 drug[5] = str(new\_quantity)  
 updated = True  
  
 updated\_drugs.append(drug)  
  
 if updated:  
 with open(file\_path, 'w') as file:  
 for drug in updated\_drugs:  
 file.write(','.join(drug) + '\n')  
  
 if operation == '出库':  
 print(f'药品数量更新成功！药品编号为 {drug\_number}，批次号为 {drug\_uuid} 的数量变更为 {new\_quantity}.')  
 elif operation == '入库':  
 print(f'药品数量更新成功！药品编号为 {drug\_number}，批次号为 {drug\_uuid} 的数量变更为 {new\_quantity}.')  
 else:  
 print(f'未找到匹配的药品记录。请检查药品编号和批次号是否正确。')

**(9)子程序八**

def drug\_date\_management():  
 file\_path = '药品清单.txt'  
 drugs = []  
  
 try:  
 with open(file\_path, 'r') as file:  
 for line in file:  
 drug\_info = line.strip().split(',')  
 drugs.append(drug\_info)  
 except FileNotFoundError:  
 print('药品信息文件不存在。')  
 return  
  
 while True:  
 print('药品生产日期和其他管理菜单：')  
 print('1. 显示临期药品')  
 print('2. 显示过期药品')  
 print('0. 返回上一级菜单')  
 choice = input('请输入数字选择功能：')  
  
 if choice == '1':  
 today = datetime.datetime.now().date()  
 for drug in drugs:  
 expiry\_date = datetime.datetime.strptime(drug[3], '%Y-%m-%d').date()  
 days\_left = (expiry\_date - today).days  
 if days\_left < 45 and days\_left > 0:  
 print(f'药品编号: {drug[1]}')  
 print(f'药品名称: {drug[0]}')  
 print(f'生产日期: {drug[2]}')  
 print(f'有效期: {drug[3]}')  
 print('层数', drug[7])  
 print('种类', drug[8])  
 print(f'剩余天数: {days\_left}')  
 print('---')  
 elif choice == '2':  
 today = datetime.datetime.now().date()  
 for drug in drugs:  
 expiry\_date = datetime.datetime.strptime(drug[3], '%Y-%m-%d').date()  
 days\_left = (expiry\_date - today).days  
 if days\_left < 0:  
 print(f'药品编号: {drug[1]}')  
 print(f'药品名称: {drug[0]}')  
 print(f'生产日期: {drug[2]}')  
 print(f'有效期: {drug[3]}')  
 print('层数', drug[7])  
 print('种类', drug[8])  
 print('---')  
 elif choice == '0':  
 return  
 else:  
 print('无效的选项。')

**(10)子程序九**

def drug\_information\_query():  
 file\_path = '药品清单.txt'  
 drugs = []  
  
 try:  
 with open(file\_path, 'r') as file:  
 for line in file:  
 drug\_info = line.strip().split(',')  
 drugs.append(drug\_info)  
 except FileNotFoundError:  
 print('药品信息文件不存在。')  
 return  
  
 drug\_number = input('请输入要查询的药品编号：')  
  
 found\_drugs = [] # 用于存储符合条件的药品信息  
  
 for drug in drugs:  
 if drug[1] == drug\_number:  
 found\_drugs.append(drug)  
  
 if found\_drugs:  
 print(f'药品编号为 {drug\_number} 的药品信息：')  
 for drug in found\_drugs:  
 print('药品名称:', drug[0])  
 print('药品编号:', drug[1])  
 print('生产日期:', drug[2])  
 print('有效期:', drug[3])  
 print('批次号:', drug[4])  
 print('数量:', drug[5])  
 print('货架号',drug[6])  
 print('层数',drug[7])  
 print('种类',drug[8])  
 else:  
 print(f'未找到药品编号为 {drug\_number} 的药品。')

1. **子程序十**

def check\_low\_quantity\_drugs():  
 file\_path = '药品清单.txt'  
 low\_quantity\_drugs = []  
  
 try:  
 with open(file\_path, 'r') as file:  
 for line in file:  
 drug\_info = line.strip().split(',')  
 if len(drug\_info) > 5:  
 drug\_quantity = int(drug\_info[5])  
 if drug\_quantity <= 50:  
 low\_quantity\_drugs.append(drug\_info)  
 except FileNotFoundError:  
 print('药品信息文件不存在。')  
 return  
  
 if len(low\_quantity\_drugs) > 0:  
 with open('需要补充药品.txt', 'w') as file:  
 for drug in low\_quantity\_drugs:  
 file.write(','.join(drug) + '\n')  
  
 print('需要补充的药品信息已写入文件 "需要补充药品.txt"。')  
 else:  
 print('当前没有需要补充的药品。')

1. **主程序十一**

def drug\_category\_query():  
 file\_path = '药品清单.txt'  
 drugs = []  
  
 try:  
 with open(file\_path, 'r') as file:  
 for line in file:  
 drug\_info = line.strip().split(',')  
 drugs.append(drug\_info)  
 except FileNotFoundError:  
 print('药品信息文件不存在。')  
 return  
  
 category\_mapping = {  
 '1': '退烧药',  
 '2': '消炎药',  
 '3': '感冒药',  
 '4': '维生素',  
  
 }  
  
 show\_menu()  
  
 category\_input = input("请输入药品种类编号：")  
  
 if category\_input in category\_mapping:  
 category = category\_mapping[category\_input]  
  
 found\_drugs = [] # 用于存储符合条件的药品信息  
  
 for drug in drugs:  
 if len(drug) > 6 and category in drug[6:]:  
 found\_drugs.append(drug)  
  
 if found\_drugs:  
 print(f'种类为 {category} 的药品信息：')  
 for drug in found\_drugs:  
 print('药品名称:', drug[0])  
 print('药品编号:', drug[1])  
 print('生产日期:', drug[2])  
 print('有效期:', drug[3])  
 print('批次号:', drug[4])  
 print('数量:', drug[5])  
 print('货架号', drug[6])  
 print('层数', drug[7])  
 print('---')  
 else:  
 print(f'未找到种类为 {category} 的药品。')  
 else:  
 print("输入的药品种类编号无效")  
def show\_menu():  
 print("请选择药品种类：")  
 print("1. 退烧药")  
 print("2. 消炎药")  
 print("3. 感冒药")  
 print("4. 维生素")

1. **主程序十二**

def show\_file\_content(file\_path):  
 try:  
 with open(file\_path, 'r') as file:  
 content = [line.strip().split(',') for line in file.readlines()]  
  
 # 计算每个字段的最大长度  
 max\_lengths = [max(len(item) for item in column) for column in zip(\*content)]  
  
 # 对于数字字段，使用固定的宽度  
 for i in range(len(max\_lengths)):  
 if i == 1 or i == 5 or i == 6 or i == 7: # 药品编号、库存、最小库存、最大库存  
 max\_lengths[i] = max(max\_lengths[i], 10)  
  
 # 打印内容  
 print('药品名称 | 药品编号 | 生产日期 | 有效期 | 批次号 |库存 |货架 |层数 |药品类别')  
  
 for row in content:  
 row\_line = ' | '.join('{:<{}}'.format(item, max\_length) for item, max\_length in zip(row, max\_lengths))  
 print(row\_line)  
 except FileNotFoundError:  
 print(f'文件 {file\_path} 不存在。')

1. **主程序十三**

def show\_file():  
 while True:  
 print("1. 展示药品清单文件")  
 print("2. 展示需要补充药品清单")  
 print("0. 返回上级菜单")  
 choice = int(input("请选择你要展示的文件: "))  
  
 if choice == 1:  
 show\_file\_content('药品清单.txt')  
 elif choice == 2:  
 show\_file\_content('需要补充药品.txt')  
 elif choice == 0:  
 break # 跳出循环返回到上级菜单  
 else:  
 print("无效的选择")

**（n）主程序**

login()