**python期末报告要求：**

# 系统功能实现

系统功能主题不限。系统设计必须体现模块化设计思想，建议采用面向对象设计思想进行开发，代码规范，条理清晰。该系统必须包括但不限于登陆功能，存盘功能，不少于三个系统主功能，可以有退出功能。具体功能要求参考如下：

登陆功能：根据身份证号，手机号，姓名等身份信息进行登录设计。

退出功能：设计退出限制，如必须输入密码等。

存盘功能：执行存盘功能后，程序执行的数据会写入本地文件。

主功能：如图书管理系统中管理员对图书的增加、删除、修改和查询功能等。

可以包含扩展功能：比如图书管理系统中普通用户对图书的借阅、归还、查询等。

**凡涉及管理员账号，一律设计登录名为：root，密码为：123**

系统主题推荐：\*\*仓库管理系统，教务系统，图书管理系统，医院监护系统，\*\*游戏系统等（仅供参考）

# 另：

除以上管理类系统的设计外，亦可以设计一个图形可视化的登录管理界面，有用户登录、注册等功能，能够从本地文件中读取用户列表进行登录的验证，注册能够通过对本地用户文件的修改增加新的用户信息，能够进行随机验证码的验证，在界面右下角显示作者信息，包括学号后三位和姓名。

如下图：



# 提交要求

各自完成的作品，需要提交文档和项目报告，完成的设计报告可参考**附件1：报告模板参考**。

# 报告提交时间

2021年12月27日上课前，各班负责人收齐报告，按照学号顺序排好上交。

# 附件1：报告模板参考

**《Python期末程序设计报告》**

班级：

姓名：

学号：

成绩：

2021年12月

山东外事职业大学信息与控制工程学院

**目 录**

[1 需求分析 1](#_Toc469256348)

[1.1系统背景介绍 1](#_Toc469256349)

[1.2功能需求分析 1](#_Toc469256350)

[2 总体设计 1](#_Toc469256351)

[2.1系统功能结构 1](#_Toc469256352)

[2.2系统流程图 1](#_Toc469256353)

[2.3开发及运行环境 1](#_Toc469256354)

[3 详细设计 1](#_Toc469256355)

[4 代码实现 1](#_Toc469256356)

[4.1业务模型图 1](#_Toc469256357)

[4.2程序运行截图 2](#_Toc469256358)

[5 测试 2](#_Toc469256359)

[6 总结 2](#_Toc469256360)

# 1 需求分析

1.1系统背景介绍

仓库管理系统对于学校的管理者来说十分重要， 学校基础设备的管理离不开仓库系统，一个好的仓库管理系统方便管理员的设备资源管理，节省管理费用，减少不必要的失误。

1.2功能需求分析

**需求：**

1. 登陆界面，必须输入正确的账号和密码才能登陆
2. 提供入库，出库，添加设备等功能
3. 通过编号，类型等查询相应设备
4. 使用sqlite将数据存储在数据库文本当中

分析：使用sqlite和sql语句执行相应的增加，查询，删除等工作

定义login函数来判断用户是否合法。使用split等方法解析用户输入，并放入相应的变量，最后格式化插入sql语句中，使用python与sqlite的接口执行sql语句的增删改查和统计等工作。

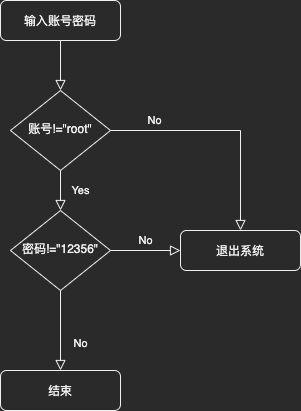
# 2 总体设计

2.1系统功能结构

本系统主要完成对信息学院的各种设备的管理，包括设备的入库、设备的出库、设备的查询、仓库设备总价格统计等功能，通过SQLite数据库保存到数据库文件。

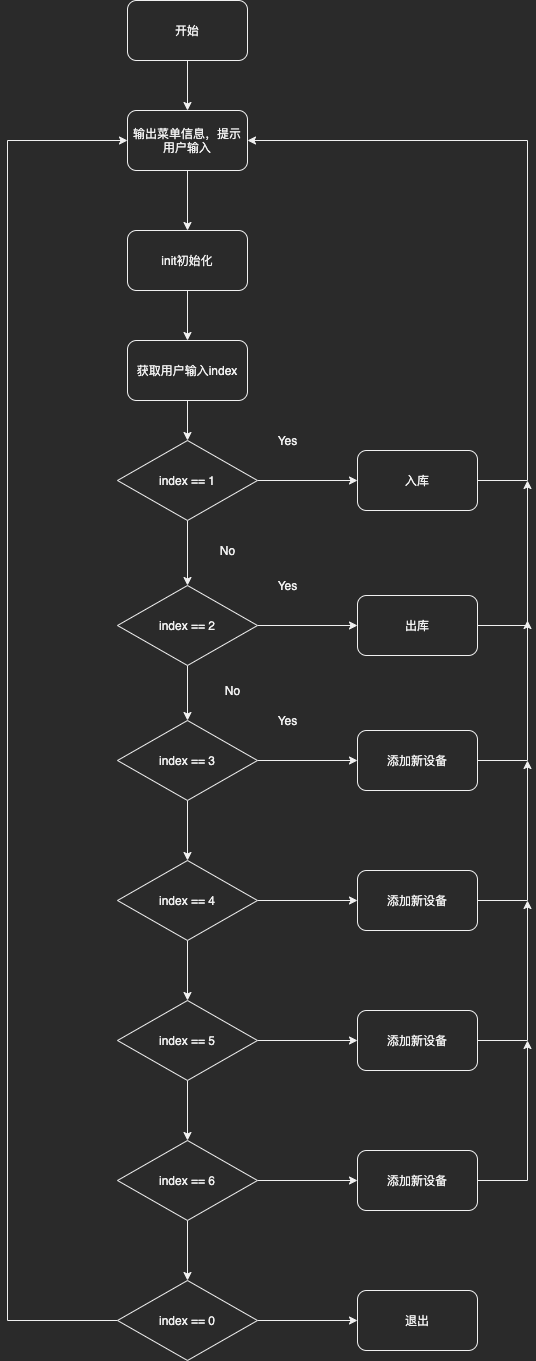
2.2系统流程图

（1）login函数



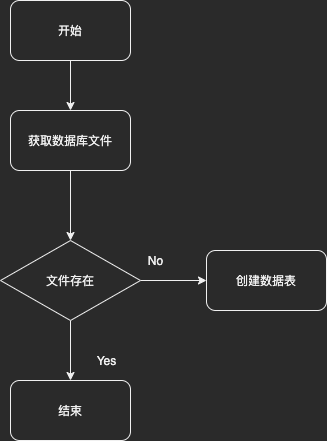
（图2-1）

1. main函数



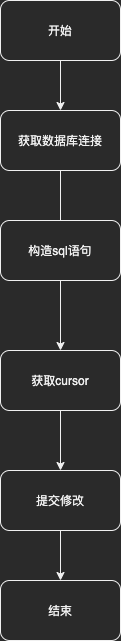
（图2-2）

（3）init函数



（图2-3）

（4）增删改查函数



（2-4）

2.3开发及运行环境

本系统是基于 64 位 windows10专业版，硬件基于 cpu i5-7200m, 内存 4GB。程序将基于python3.5，运用pycharm完成整体开发。

# 3 详细设计

本系统主要使用sqlite维护信息学院的设备数据，分别提供增加，删除，修改，查询设备的接口。首先执行Login函数使用if语句简单判断用户名和密码是否合法，如果非法则退出系统。使用init函数进行程序初始化，该函数检查是否已经有相应数据库文件存在，如果存在则读入数据库文件信息，如果不存在则创建新的数据表，并且保存到新的数据库文件，之后的一系列操作都将数据存放至该数据库。开始时使用while true进行无限循环进行驱动，不断输出菜单列表促使用户输入选项。通过if else等语句进行判断用户的输入从而进行不同的函数操作。最后关闭数据库连接，进行一系列程序结束操作。

# 4 代码实现

4.1代码

# 插入新元件

def insertInStorage(conn, data):

cur = conn.cursor()

insert = 'insert into electronicComponents ' \

'values (?,?,?,?,?,?,?,?,?)'

cur.execute(insert, data)

conn.commit()

cur.close()

# 初始化

def init(db\_filename):

path = pathlib.Path(db\_filename)

if (not path.exists()):

try:

conn = sqlite3.connect(db\_filename)

except sqlite3.Error as e:

print(e)

return

cur = conn.cursor()

creatElectronicComponentsTable = 'create table electronicComponents(' \

'no integer,' \

'name text,' \

'type text,' \

'resistance integer,' \

'nominalVoltage integer,' \

'nominalCurrent integer,' \

'price integer,' \

'quantity integer,' \

'info text,' \

'primary key(no))'

createRecordTable = 'create table record(' \

'rno integer,' \

'cno integer, ' \

'quantity integer,' \

'date datetime, ' \

'primary key(rno autoincrement),' \

'foreign key(cno) references electronicComponents(no))'

creatIndex = 'create index time on record(date)'

cur.execute(creatElectronicComponentsTable)

cur.execute(createRecordTable)

cur.execute(creatIndex)

# loadData(conn)

cur.close()

else:

conn = sqlite3.connect(db\_filename)

return conn

# 跟新库存

def updateStorage(conn, data):

cur = conn.cursor()

update = 'update electronicComponents set quantity = quantity + ? where no = ?'

cur.execute(update, data)

conn.commit()

cur.close()

# 写入销售记录

def insertRecord(conn, data):

cur = conn.cursor()

insert = 'insert into record(cno,quantity,date) ' \

'values (?,?,datetime("now"))'

cur.execute(insert, data)

conn.commit()

cur.close()

# 检查库存是否充足

def checkInventory(conn, data):

cur = conn.cursor()

select = 'select quantity ' \

'from electronicComponents ' \

'where no = ? '

cur.execute(select, data)

res = cur.fetchall()

conn.commit()

cur.close()

if (res[0][0] <= inventoryAlertValue):

return False

else:

return True

# 更新库存

def update(conn, data):

updateStorage(conn, (data[1], data[0]))

if (data[1] < 0):

insertRecord(conn, (data[0], -data[1]))

res = checkInventory(conn, (data[0],))

return res

# 根据类型查询功能

def getByType(conn, data):

cur = conn.cursor()

select = 'select \* ' \

'from electronicComponents ' \

'where type = ?'

cur.execute(select, data)

res = cur.fetchall()

cur.close()

return res

# 根据编号查询信息

def getByNo(conn, data):

cur = conn.cursor()

select = 'select \* ' \

'from electronicComponents ' \

'where no = ? '

cur.execute(select, data)

res = cur.fetchall()

cur.close()

return res

# 获取库存总价值

def getInventoryValue(conn):

cur = conn.cursor()

select = 'select sum(quantity\*price) ' \

'from electronicComponents '

cur.execute(select)

res = cur.fetchall()

cur.close()

return res

def login():

username = input("请输入管理员账号：")

passwd = input("请输入管理员密码：")

if username != "root" and passwd != "123":

print("账户或密码错误")

exit(-1)

login()

conn = init('xinyuancangku.db')

while (True):

print('-' \* 10 + '信院管理系统' + '-' \* 10)

print('1.入库')

print('2.出库')

print('3.添加新设备')

print('4.按照类型查看功能')

print('5.按照编号查看信息')

print('6.仓库设备总价值')

print('0.退出')

print('请输入选择的功能编号：')

idex = input()

if idex == '1':

os.system('cls')

print('请依次输入设备编号和数量，使用空格隔开：')

data = input().split()

data[0] = int(data[0])

data[1] = int(data[1])

data = tuple(data)

update(conn, data)

elif idex == '2':

os.system('cls')

print('请依次输入设备件编号和数量，使用空格隔开：')

data = input().split()

data[0] = int(data[0])

data[1] = -int(data[1])

data = tuple(data)

res = update(conn, data)

if res == False:

print('库存不足')

print('按回车返回主界面')

input()

elif idex == '3':

os.system('cls')

print('请依次输入设备编号(数字)，名字，类型，电阻(数字)，额定电压(数字)，额定电流(数字)，价格(数字)，数量(数字)，说明，使用空格隔开：')

data = input().split()

for i in range(4, 8):

data[i] = int(data[i])

data = tuple(data)

insertInStorage(conn, data)

elif idex == '4':

os.system('cls')

print('请输入设备类型：')

data = input()

data = (data,)

res = getByType(conn, data)

print('编号' + '\t' + '功能')

for row in res:

print(row[1] + '\t' + row[-1])

print('按回车返回主界面')

input()

os.system('cls')

elif idex == '5':

os.system('cls')

print('请输入设备编号：')

data = int(input())

data = (data,)

res = getByNo(conn, data)

print(

'编号' + '\t' + '名字' + '\t' + '类型' + '\t' + '电阻' + '\t' + '额定电压' + '\t' + '额定电流' + '\t' + '价格' + '\t' + '数量' + '\t' + '信息')

for row in res:

for item in row:

print(item, end='')

print('\t', end='')

print()

print('按回车返回主界面')

input()

os.system('cls')

elif idex == '6':

res = getInventoryValue(conn)

print('总价值为：', end='')

print(res[0][0])

print('按回车返回主界面')

input()

os.system('cls')

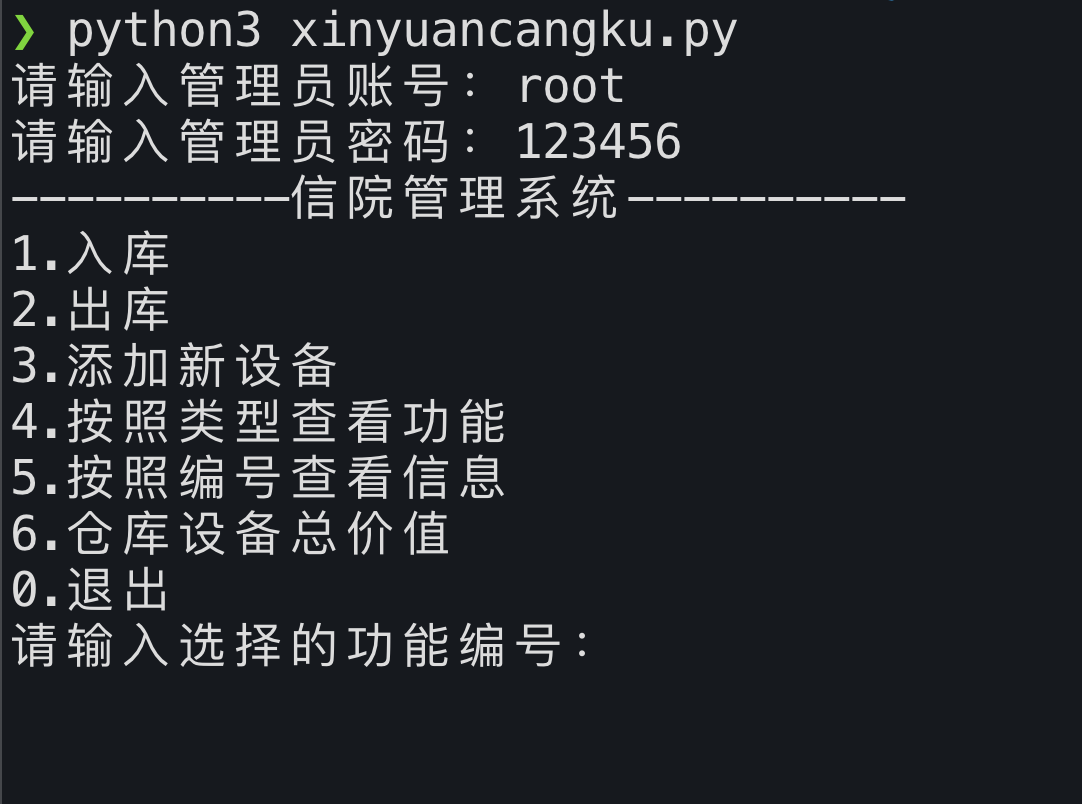
elif idex == '0':

break

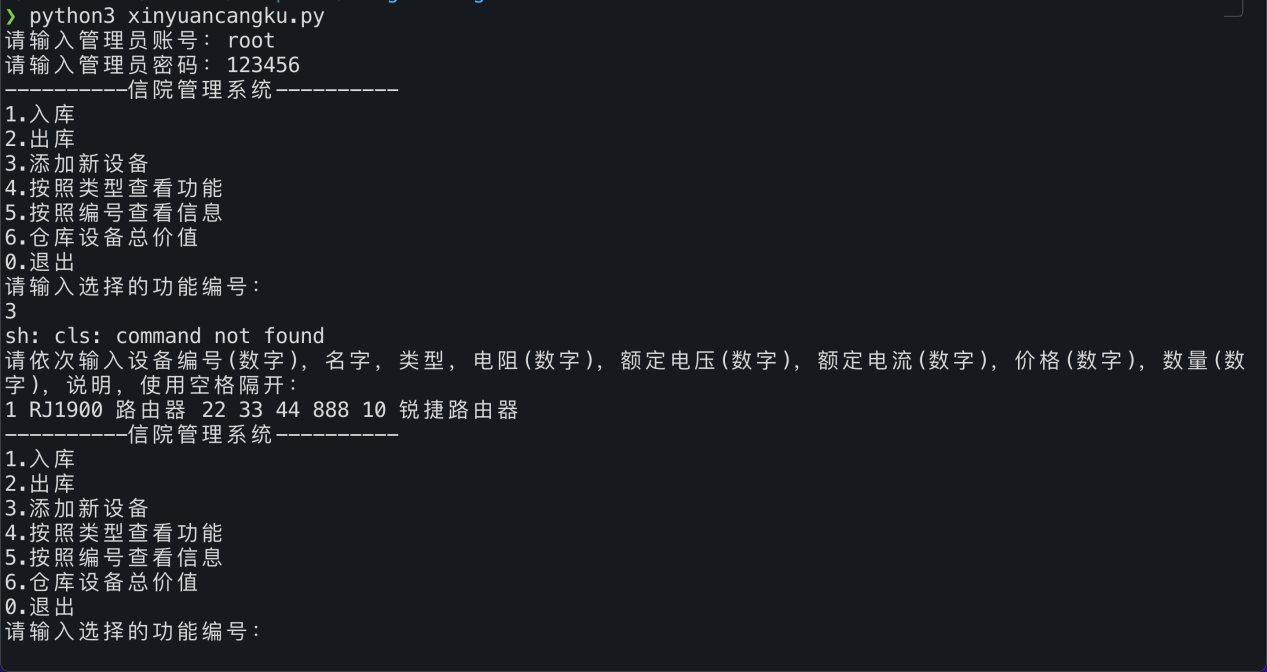
conn.close()

4.2程序运行截图

登录界面：



添加新设备：



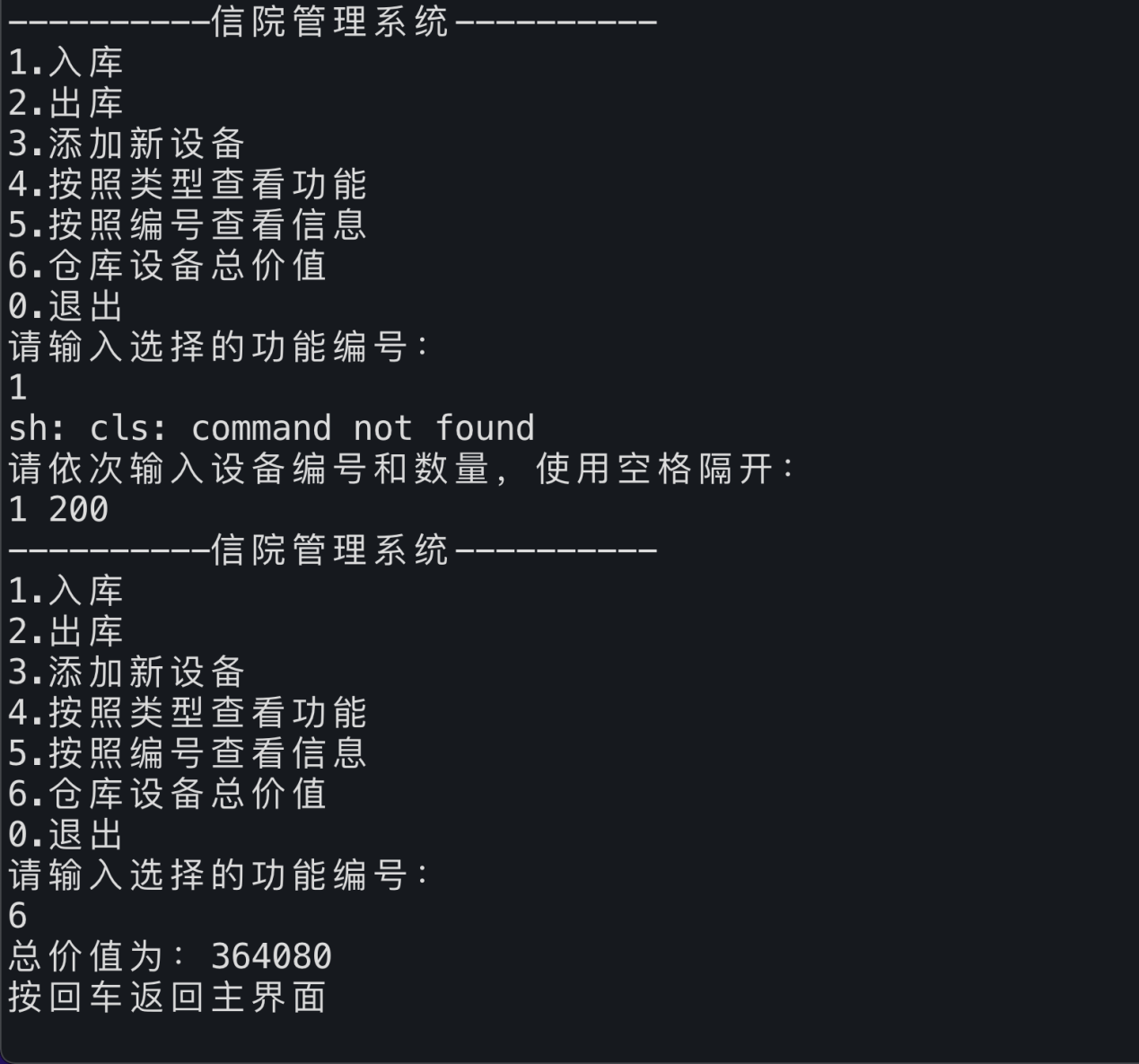
查询成绩：

（1）学生查询

查看设备总价值:

## Screen Shot 2021-12-21 at 8.49.27 PM

入库200台锐捷路由器后的仓库总资产：



出库100台路由器:

## Screen Shot 2021-12-21 at 8.55.00 PM

# 5 测试

黑盒测试也称功能测试，它是通过测试来检测每个功能是否都能正常使用。在测试中，把程序看作一个不能打开的黑盒子， 在完全不考虑程序内部结构和内部特性的情况下， 在程序接口进行测试， 它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试着眼于程序外部结构， 不考虑内部逻辑结构， 主要针对软件界面和软件功能进行测试。黑盒测试主要发现以下类型错误 :(1)功能不对或遗漏 ,(2)接口错误 ,(3)数据结构访问错误 ,(4)性能错误 ,(5)初始化盒终止错误。

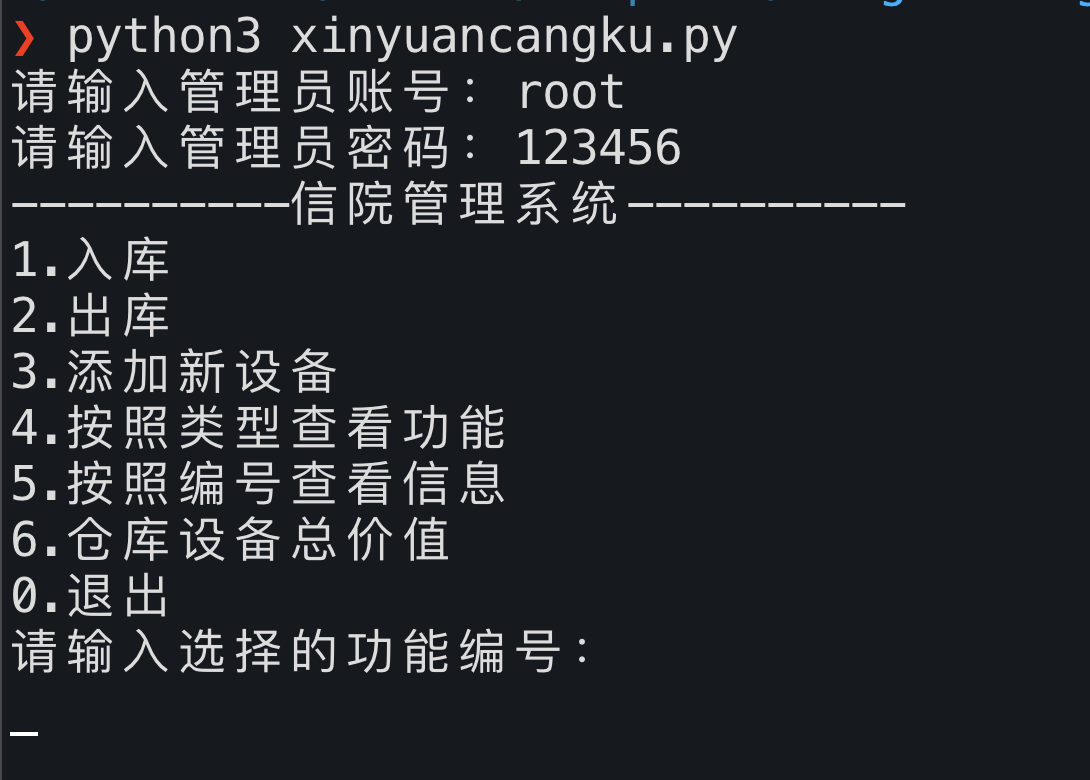
学生用户查询模块的测试中， ①有效输入有：单击主菜单下拉菜单的查询功能按键②有效输出为：弹出选择功能信息的对应的消息界面。

教师课程查询管理模块的测试中：①有效输入有：单击主菜单下拉菜单的查询与管理功能按键， ②有效输出为：弹出选择功能信息的对应的消息界面。

管理员管理模块的测试中①有效输入有：单击主菜单下拉菜单的各数据的添加按键，②有效输出为：弹出选择功能信息的对应的消息界面。

选择对用户登录模块进行测试， 测试用例为：①选仓库管理员登录， 输

入root，密码输入 123456，测试结果如下图：



测试结果与测试预期相符

# 6 总结

Python课程大报告的这次实验让我了解到一个完整的系统设计及实现流程， 通过对学生成绩管理信息系统的分析、设计、编写、测试、管理等过程，使我更加理解学生信息管理系统的生产流程，了解到对于一个系统从分析到最后完成出产的过程是非常复杂的，不是仅仅只有编写代码才是重要的，和相关文档工作也是十分重要的，甚至重要性和完成难度要比一个完整的代码编写还要高。所以本次实验对我写代码的风格、相关文档的分析工作，及测试能力有了很多的提高，对于相关的知识也有了更深的理解。

**期末考查结果评定**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 评分项目 | 分值 | 得分 | | 报告条理清晰，内容详实，体会深刻 | 40 |  | | 报告格式符合规范 | 10 |  | | 系统功能实现符合要求 | 40 |  | | 界面美观，功能有扩充 | 10 |  |   评语： |
| 成绩： |
| 指导教师签字：  评定日期： 年 月 日 |
|