|  |
| --- |
| **一、设计任务及要求：**  **设计任务：住院病人管理系统的设计与实现**  **要 求：**  设计一个住院病人管理系统，主要功能包括管理员的注册和登录，管理员登陆后对病人信息的录入和查询、医生信息的录入及查询 |
| **二、指导教师评语：**   |  |  | | --- | --- | |  | 系统功能完善；回答问题正确、熟练；论文内容设计合理，文理通顺，图表完备、正确；平时表现良好。 | |  | 系统功能基本实现；回答问题基本正确；论文内容设计合理，格式正确；平时表现良好。 | |  | 系统大部分功能实现；回答问题大部分正确；论文内容设计合理，格式正确；平时表现良好。 | |  | 系统功能不全；回答问题错误多；论文内容设计合理，格式正确；平时表现一般。 | |  | 系统不能正常运行；回答问题错误过多或者未答辩；论文未交或者质量太差；平时缺勤较多。 | |
| **三、成绩**  优秀 良好 中等 及格 不及格    指导教师签名：    年 月 日 |

**住院病人管理系统的开发**

1选题的意义

医院中众多的医生以及病人信息，通过传统的文件形式已经远远的满足不了人们的需求，应运而生的各类管理信息系统为医生和病人的信息管理提供了方便

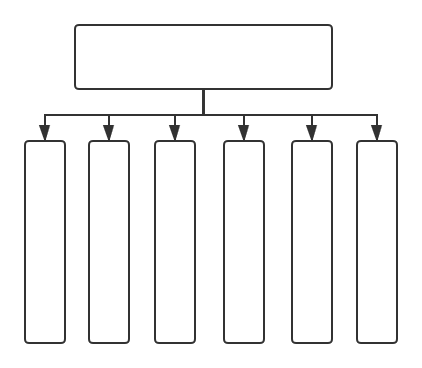
2 系统功能需求分析

2.1系统概述

我们设计的住院病人系统共有6个功能，所涉及到的功能有管理员的注册和登录，病人信息的增加和查询、医生信息的录入和医生相关信息的查询等。

2.2系统的组成及各模块的功能

住院病人管理系统



医生信息查询

住院病人信息录入

医生信息录入

住院病人信息查询

管理员注册

管理员注册

图 1 总体模块图

2.3系统的运行环境

系统的运行环境为windows10平台的pycharm

Python的版本为 3.7

3系统设计

3.1总体结构设计

将程序设计成义菜单的方式工作，并利用python语言中常用的for循环、if函数以及数据库中sql来完成功能。主要如下：

main() 该函数为主要的入口函数

record\_doctor\_data() 该函数用来增加医生的信息

select\_doctor\_data() 该函数用来查询医生的相关信息

select\_patient\_data() 该函数用来增加病人的相关信息

select\_patient\_data() 该函数用来查询病人的相关信息

3.2模块函数设计

主要是医生的管理模块，包括医生信息的增加以及医生信息的查询。

1. 医生信息的增加：通过输入医生的代码、姓名、性别、职称来增加医生的信息，然后根据sql关键字 insert来插入到数据库。
2. 医生信息的查询：通过输入医生代码或者医生名称来查询医生的信息以及医生所看的病人的信息，通过sql关键字 select来查询

4系统实现

4.1菜单和主函数的设计与实现

def main():

while True:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\* 欢迎使用住院病人管理系统 \*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\* 请选择功能（序号）： \*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\* 1.登录 \*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\* 2.注册 \*\*\*\*\*\*\*\*\*\*") print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

fun\_num = input("请输入功能序号：")

print(type(fun\_num))

if int(fun\_num) == 1:

login()

#登陆后跳出循环

break

elif int(fun\_num) == 2:

register()

else:

print("请输入合法功能序号")

def main\_fun():

while True:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\* 欢迎使用住院病人管理系统 \*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\* 请选择功能（序号）： \*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\* 1.病人信息录入 \*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\* 2.病人信息查询 \*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\* 3.医生信息录入 \*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\* 4.医生信息查询 \*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\* 5.退出 \*\*\*\*\*\*\*\*\*\*") print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

fun\_number = int(input("请输入功能序号"))

#实例化两个类，分别是病人模块和医生模块

p = Patient\_manager()

d = Doctor\_manager()

if fun\_number == 1:

#增加病人记录

p.record\_patient\_data()

elif fun\_number == 2:

#查询病人记录

p.select\_patient\_data()

elif fun\_number == 3:

#增加医生信息

d.record\_doctor\_data()

elif fun\_number == 4:

#查询医生信息

d.select\_doctor\_data()

else:

print("退出系统")

sys.exit(0)

4.2其他模块的设计与实现

登录功能

def login():

print("开始进行登陆。。。。")

username = input("请输入用户名：")

password = input("请输入密码：")

sql = "select 用户密码 from user where 用户名 = " + "'" +username +"'"

cursor.execute(sql)

#查询用户

result = cursor.fetchone()

if result:

while result[0] != password:

password = input("密码错误，请重新输入：")

main\_fun()

else:

print("用户不存在！，请注册：")

register()

注册功能：

def register():

user\_number = input("请输入用户编号：（小于10个字符）")

username = input("请输入用户名：（小于8个字符）")

password = input("请输入密码：（小于20个字符）")

password\_again = input("请再次输入密码：（小于20个字符）")

# pycharm中无法使用getpass 可以脚本启动时更换

# password = getpass.getpass("请输入密码：（小于20个字符）")

# password\_again = getpass.getpass("请再次输入密码：（小于20个字符）")

user\_type = input("请输入用户类型：（小于10个字符）")

#判断两次输入的密码是不是一致

while(password != password\_again):

print("两次输入的密码不一致！")

user\_number = input("请输入用户编号：（小于10个字符）")

username = input("请输入用户名：（小于8个字符）")

# password = getpass.getpass("请输入密码：（小于20个字符）")

# password\_again = getpass.getpass("请再次输入密码：（小于20个字符）")

password = input("请输入密码：（小于20个字符）")

password\_again = input("请再次输入密码：（小于20个字符）")

user\_type = input("请输入用户类型：（小于10个字符）")

#注册信息sql将数据插入到数据库

sql = "insert into user (用户编号,用户名,用户密码,用户类型) values ('"+user\_number+"','"+username+"','"+password+"','"+user\_type+"')"

try:

#执行事务

cursor.execute(sql)

#提交事务

db.commit()

print("注册成功！")

#注册成功后转向登录

login()

except Exception as e:

#如果数据有问题出现异常，事务回滚

db.rollback()

print("注册失败" + str(e))

增加医生信息：

def record\_doctor\_data(self):

print("增加医生记录")

doctor\_number = input("请输入医生代码:")

doctor\_name = input("请输入医生姓名:")

doctor\_sex = input("请输入医生性别:")

doctor\_position = input("请输入医生职称:")

sql1 = "insert into zyys (医生代码,姓名,性别,职称) " \

"values ('"+doctor\_number+"','"+doctor\_name+"','"+doctor\_sex+"','"+doctor\_position+"')"

try:

cursor.execute(sql1)

db.commit()

print("sql1信息插入成功")

except Exception as e:

db.rollback()

print("sql1信息插入失败,异常信息：" + str(e))

查询医生信息：

def select\_doctor\_data(self):

doctor\_name = ""

doctor\_number = ""

sql1 = ""

while True:

print("请选择查询方式: 1.通过医生姓名查找 2.通过医生代码查询")

num = int(input("请输入序号"))

if num == 1:

doctor\_name = input("请输入医生姓名:")

break

elif num == 2:

doctor\_number = str(input("请输入医生代码:"))

break

else:

print("输入有误！")

if doctor\_name:

sql1 = "select zybr.住院号,zybr.姓名,zybr.性别,年龄,民族,入院日期,诊断 " \

"from zyys,zysf,zybr " \

"where zyys.医生代码 = zysf.医生代码 and zysf.住院号 = zybr.住院号 and zyys.姓名 = '" + doctor\_name +"'"

if doctor\_number:

sql1 = "select zybr.住院号,zybr.姓名,zybr.性别,年龄,民族,入院日期,诊断 " \

"from zysf,zybr " \

"where zysf.住院号 = zybr.住院号 and 医生代码 = '" + doctor\_number +"'"

if sql1:

cursor.execute(sql1)

db.commit()

print("sql1查询成功！")

print("姓名 性别 年龄 民族 入院日期 住院号 诊断")

results = cursor.fetchall()

for row in results:

patient\_name = row[1]

patient\_sex = row[2]

patient\_age = row[3]

patient\_nation = row[4]

doctor\_date =row[5].strftime("%Y-%m-%d %H:%M:%S")

patient\_hospital\_number = row[0]

sop = row[6]

print(patient\_name," "+patient\_sex," "+patient\_age," "+patient\_nation," "+doctor\_date," "+patient\_hospital\_number," "+sop)

5设计体会

5.1总结

通过设计该系统，对学习的python课程有了熟练的运用，对数据库设计和使用也有了进一步的了解。

5.2存在的主要问题

对数据库的使用不是特别的熟练，还需要多多练习。

参考文献

1. 刘宇宙.《Python 3.5 从零开始学》,北京:清华大学出版社
2. 马克·卢茨.《Python学习手册》,机械工业出版社
3. 明日科技.《Python项目案例开发集锦》,吉林大学出版社
4. 张工厂.《MySql 5.7从入门到精通》,清华大学出版社
5. Paul DuBois.《MySql经典实例》,中国电力出版社

关于格式的说明：

正文（宋体，小四号，行间距固定值20磅）

正文包括选题的意义、系统功能需求分析、系统设计、系统实现及总结部分。

1. 选题的意义：

选择做这样一个软件的用处或者好处。

2.系统功能需求分析

2.1系统概述

2.2系统的组成及各模块的功能

2.3系统的运行环境

3.系统设计

3.1总体结构设计

3.2函数功能描述

**插图的说明：**1、先插入图片，居中

2、图题要求：图题位于图的下方，居中，5号，黑体，第1张图，然后是第2张图，图题是“图2 登录界面”

3、设置如下图所示。

**图序采用图1，图2的字样，后空1中文字符。**

**图序和图题用五号黑体字，居中放于插图下方。**

**图中的文字，宋体，五号。**

图序和表序是相互独立的。

系统登陆

图书销售管理系统

图书管理

系统管理

购买图书

退出

图1 总体模块图

**课程设计（论文）页面要求：须用A4(210×297mm)标准、一律采用单面打印；页边距按以下标准设置：上边距(天头)为：3.2cm;下边距（地脚）2.8cm;左边距3cm;右边距为：2.5cm;左边装订。**

小学期日程安排：