**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

**WORK-INTEGRATED LEARNING PROGRAMMES DIVISION**

**BITS-WIPRO Collaborative Programme:**

**M. Tech in Software Engineering 2017-2018**

 

# **Assignment on Real Time Operating System**

**Faculty Name:**

 **Submitted By:**

DATE:

QUESTION:

1. Star Hospital wants to design a telemedicine setup for their hospital and also plan to do surgeries also in the remote manner by taking advice from experts at different locations. The hospital has branches in Kolkata, Delhi, Mumbai, Pune and Chennai. Suppose you have been approached to design a real time system for them, how would you  implement the same. Explain the approach  with a block diagram. Highlight the different tasks you would accomplish through the design. What would be the nature of the scheduler (Substantiate your answer with appropriate justification for the same) and explain the simulation with an example.
2. A system contains the following five periodic tasks. The tasks are scheduled rate monotonically.

 T1 = (6, 3, [X;2])

 T2 = (20, 5, [Y;2])

 T3 = (200, 5, [X;3 [Z;1]])

 T4 = (210, 6, [Z;5 [Y;4]])

 Compare the schedulability of the system when the priority ceiling protocol is used versus the NPCS protocol.

 Use a simulation tool like Cheddar and provide the screenshots

**Answer:**

**Python Script**

window2.py

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

import tkinter

import sqlite3

import tkinter.messagebox

from PATDELSU import P\_display

from PATDELSU import D\_display

from PATDELSU import P\_UPDATE

from RooMT import Room\_all

from BILLING import BILLING

from employee\_reg import emp\_screen

from appointment import appo

conn=sqlite3.connect("MDBA.db")

print("DATABASE CONNECTION SUCCESSFUL")

#variables

root1=None

rootp=None

pat\_ID=None

pat\_name=None

pat\_dob=None

pat\_address=None

pat\_sex=None

pat\_BG=None

pat\_email=None

pat\_contact=None

pat\_contactalt=None

pat\_CT=None

#EXIT for MENU

def ex():

 global root1

 root1.destroy()

#MENU BUTTONS

def menu():

 global root1,button1,button2,button3,button4,button5,m,button6

 root1=tkinter.Tk()

 root1.geometry("280x350")

 root1.title("MAIN MENU")

 m=tkinter.Label(root1,text="MENU",font='Times 16 bold italic',fg='grey')

 button1=tkinter.Button(root1,text="1.PATIENT REGISTRATION",command=PAT,bg='light blue',fg='black')

 button2 = tkinter.Button(root1, text="2.ROOM ALLOCATION",bg='light green',fg='black',command=Room\_all)

 button3 = tkinter.Button(root1, text="3.EMPLOYEE REGISTRATION",bg='light blue',fg='black',command=emp\_screen)

 button4 = tkinter.Button(root1, text="4.BOOK APPOINTMENT",bg='light green',fg='black',command=appo)

 button5 = tkinter.Button(root1, text="5.PATIENT BILL",bg='light blue',fg='black',command=BILLING)

 button6 = tkinter.Button(root1, text="6.EXIT",command=ex,bg='light green',fg='black')

 m.place(x=75,y=5)

 button1.pack(side=tkinter.TOP)

 button1.place(x=80,y=50)

 button2.pack(side=tkinter.TOP)

 button2.place(x=80,y=100)

 button3.pack(side=tkinter.TOP)

 button3.place(x=80,y=150)

 button4.pack(side=tkinter.TOP)

 button4.place(x=80, y=200)

 button5.pack(side=tkinter.TOP)

 button5.place(x=80,y=250)

 button6.pack(side=tkinter.TOP)

 button6.place(x=80,y=300)

 root1.mainloop()

p=None

#input patient form

def IN\_PAT():

 global pp1, pp2, pp3, pp4, pp5, pp6, pp7, pp8, pp9, pp10,ce1,conn

 conn=sqlite3.connect("MDBA.db")

 conn.cursor()

 pp1=pat\_ID.get()

 pp2=pat\_name.get()

 pp3=pat\_sex.get()

 pp4=pat\_BG.get()

 pp5=pat\_dob.get()

 pp6=pat\_contact.get()

 pp7=pat\_contactalt.get()

 pp8=pat\_address.get()

 pp9=pat\_CT.get()

 pp10=pat\_email.get()

 conn.execute('INSERT INTO PATIENT VALUES(?,?,?,?,?,?,?,?)',(pp1,pp2,pp3,pp4,pp5,pp8,pp9,pp10,))

 conn.execute('INSERT INTO CONTACT\_NO VALUES (?,?,?)',(pp1,pp6,pp7,))

 tkinter.messagebox.showinfo("MEDANTA DATABSE SYSTEM","DETAILS INSERTED INTO DATABASE")

 conn.commit()

#exit from patient form

def EXO():

 rootp.destroy()

#function for patient form help

def nothing():

 print("CONTACT DATABASE HEAD :921 ")

def nothing1():

 print("MADE BY BHAVIYA BATRA")

#PATIENT FORM

back=None

SEARCH=None

DELETE=None

UPDATE=None

def PAT():

 global pat\_address, pat\_BG, pat\_contact, pat\_contactalt, pat\_CT, pat\_dob, pat\_email, pat\_ID, pat\_name, pat\_sex

 global rootp,regform,id,name,dob,sex,email,ct,addr,c1,c2,bg,SUBMIT,menubar,filemenu,back,SEARCH,DELETE,UPDATE

 rootp=tkinter.Tk()

 rootp.title("MEDANTA PATIENT FORM")

 menubar=tkinter.Menu(rootp)

 filemenu=tkinter.Menu(menubar,tearoff=0)

 filemenu.add\_command(label="NEW",command=PAT)

 filemenu.add\_separator()

 filemenu.add\_command(label="EXIT", command=EXO)

 helpmenu=tkinter.Menu(menubar,tearoff=0)

 helpmenu.add\_command(label="HELP",command=nothing)

 helpmenu.add\_command(label="ABOUT",command=nothing1)

 menubar.add\_cascade(label="File", menu=filemenu)

 menubar.add\_cascade(label="Help", menu=helpmenu)

 rootp.config(menu=menubar)

 regform=tkinter.Label(rootp,text="REGISTRATION FORM",font="Arial 16 bold")

 id=tkinter.Label(rootp,text="PATIENT ID")

 pat\_ID=tkinter.Entry(rootp)

 name=tkinter.Label(rootp,text="PATIENT NAME")

 pat\_name = tkinter.Entry(rootp)

 sex=tkinter.Label(rootp,text="SEX")

 pat\_sex=tkinter.Entry(rootp)

 dob=tkinter.Label(rootp, text="DOB (YYYY-MM-DD)")

 pat\_dob=tkinter.Entry(rootp)

 bg=tkinter.Label(rootp, text="BLOOD GROUP")

 pat\_BG=tkinter.Entry(rootp)

 c1=tkinter.Label(rootp, text="CONTACT NUMBER")

 pat\_contact=tkinter.Entry(rootp)

 c2=tkinter.Label(rootp, text="ALTERNATE CONTACT")

 pat\_contactalt=tkinter.Entry(rootp)

 email=tkinter.Label(rootp, text="EMAIL")

 pat\_email = tkinter.Entry(rootp)

 ct=tkinter.Label(rootp,text="CONSULTING TEAM / DOCTOR")

 pat\_CT=tkinter.Entry(rootp)

 addr=tkinter.Label(rootp, text="ADDRESS")

 pat\_address=tkinter.Entry(rootp)

 back=tkinter.Button(rootp,text="<< BACK",command=menu)

 SEARCH=tkinter.Button(rootp,text=" SEARCH >> ",command=P\_display)

 DELETE=tkinter.Button(rootp,text=" DELETE ",command=D\_display)

 UPDATE=tkinter.Button(rootp,text=" UPDATE ",command=P\_UPDATE)

 SUBMIT=tkinter.Button(rootp,text=" SUBMIT ",command=IN\_PAT,)

 regform.pack()

 id.pack()

 pat\_ID.pack()

 name.pack()

 pat\_name.pack()

 sex.pack()

 pat\_sex.pack()

 dob.pack()

 pat\_dob.pack()

 bg.pack()

 pat\_BG.pack()

 c1.pack()

 pat\_contact.pack()

 c2.pack()

 pat\_contactalt.pack()

 email.pack()

 pat\_email.pack()

 ct.pack()

 pat\_CT.pack()

 addr.pack()

 pat\_address.pack()

 SUBMIT.pack()

 back.pack(side=tkinter.LEFT)

 UPDATE.pack(side=tkinter.LEFT)

 DELETE.pack(side=tkinter.LEFT)

 SEARCH.pack(side=tkinter.LEFT)

 rootp.mainloop()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

login.py

\_\_\_\_\_\_\_\_\_\_\_

import tkinter

from window2 import menu

#root=login page

#root1=menu

#rootp=patient form

#variables

root=None

userbox=None

passbox=None

topframe=None

bottomframe=None

frame3=None

login=None

#command for login button

def GET():

 global userbox,passbox,error

 S1=userbox.get()

 S2=passbox.get()

 if(S1=='bhaviya' and S2=='1234567'):

 menu()

 elif(S1=='sejal taneja' and S2=='btech'):

 menu()

 else:

 error=tkinter.Label(bottomframe,text="Wrong Id / Password \n TRY AGAIN",fg="red",font="bold")

 error.pack()

#LOGIN PAGE WINDOW

def Entry():

 global userbox,passbox,login,topframe,bottomframe,image\_1

 root = tkinter.Tk()

 root.geometry("280x250")

 topframe = tkinter.Frame(root)

 topframe.pack()

 bottomframe=tkinter.Frame(root)

 bottomframe.pack()

 heading = tkinter.Label(topframe, text="WELCOME TO UK NURSING",bg='white',fg='orange',font='Times 16 bold italic')

 username=tkinter.Label(topframe,text="USERNAME")

 userbox = tkinter.Entry(topframe)

 password=tkinter.Label(bottomframe,text="PASSWORD")

 passbox = tkinter.Entry(bottomframe,show="\*")

 login = tkinter.Button(bottomframe, text="LOGIN", command=GET,font="arial 8 bold")

 heading.pack()

 username.pack()

 userbox.pack()

 password.pack()

 passbox.pack()

 login.pack()

 root.title("DATABASE LOGIN")

 root.mainloop()

Entry()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

employee\_reg.py

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

import tkinter

import sqlite3

rootE=None

var=None

def inp():

 global e1,e2,e3,e4,e5,e6,e7,e8,e9,var

 e1=t1.get()

 e2=t2.get()

 e3=str(var.get())

 e4=t3.get()

 e5=lb.get(tkinter.ACTIVE)

 e6=t4.get()

 e7=t5.get()

 e8=t6.get()

 e9=t7.get()

 conn = sqlite3.connect("MDBA.db")

 conn.execute("INSERT INTO employee VALUES(?,?,?,?,?,?,?,?,?)",(e1,e2,e3,e4,e5,e6,e7,e8,e9,))

 conn.commit()

 tkinter.messagebox.showinfo("MEDANTA DATABASE SYSTEM", "EMPLOYEE DATA ADDED")

def ex():

 rootE.destroy()

def emp\_screen():

 global rootE,t1,t2,r1,r2,t3,lb,t4,t5,t6,t7,var

 rootE=tkinter.Tk()

 rootE.title("Employee registration")

 rootE.geometry('400x400')

 var = tkinter.StringVar(master=rootE)

 H=tkinter.Label(rootE,text="EMPLOYEE REGISTRATION",fg='grey',font="Arial 10 bold")

 H.place(x=50,y=20)

 l1=tkinter.Label(rootE,text="EMPLOYEE ID")

 l1.place(x="50",y="50")

 t1=tkinter.Entry(rootE)

 t1.place(x='170',y='50')

 l2 = tkinter.Label(rootE, text="EMPLOYEE NAME")

 l2.place(x=50,y=80)

 t2 = tkinter.Entry(rootE)

 t2.place(x='170', y='80')

 l3 = tkinter.Label(rootE, text="SEX")

 l3.place(x=50,y=110)

 r1 = tkinter.Radiobutton(rootE, text="MALE", variable=var, value="Male")

 r1.place(x=80, y=110)

 r2 = tkinter.Radiobutton(rootE, text="FEMALE", variable=var, value="Female")

 r2.place(x=150, y=110)

 l4 = tkinter.Label(rootE, text="AGE")

 l4.place(x=50,y=140)

 t3=tkinter.Entry(rootE)

 t3.place(x=80,y=140)

 l5 = tkinter.Label(rootE, text="EMPLOYEE TYPE")

 l5.place(x=50,y=170)

 scrollbar = tkinter.Scrollbar(rootE, width=2)

 scrollbar.place(x=260, y=140)

 lb = tkinter.Listbox(rootE, selectmode='SINGLE', exportselection=0, height=1, width=15,yscrollcommand = scrollbar.set)

 lb.insert(tkinter.END, "DOCTOR")

 lb.insert(tkinter.END, "NURSE")

 lb.insert(tkinter.END, "RECEPTIONIST")

 lb.place(x=150, y=170)

 lb.config(yscrollcommand=scrollbar.set)

 scrollbar.configure(command=lb.yview)

 l6=tkinter.Label(rootE,text="SALARY")

 l6.place(x=50,y=200)

 t4=tkinter.Entry(rootE)

 t4.place(x=110,y=200)

 l7 = tkinter.Label(rootE, text="EXPERIENCE")

 l7.place(x=50,y=230)

 t5 = tkinter.Entry(rootE)

 t5.place(x=140,y=230)

 l8 = tkinter.Label(rootE, text="MOBILE NO")

 l8.place(x=50,y=260)

 t6 = tkinter.Entry(rootE)

 t6.place(x=140,y=260)

 l9 = tkinter.Label(rootE, text="EMAIL")

 l9.place(x=50,y=290)

 t7=tkinter.Entry(rootE)

 t7.place(x=90,y=290)

 b1=tkinter.Button(rootE,text="SAVE",command=inp)

 b1.place(x=60,y=320)

 b2=tkinter.Button(rootE,text="DELETE EMPLOYEE",command=delo)

 b2.place(x=110,y=320)

 b3=tkinter.Button(rootE,text="EXIT",command=ex)

 b3.place(x=230,y=320)

 rootE.mainloop()

def delling():

 global d1,de

 de=str(d1.get())

 conn = sqlite3.connect("MDBA.db")

 p = list(conn.execute("select \* from employee where EMP\_ID=?", (de,)))

 if (len(p) != 0):

 conn.execute("DELETE from employee where EMP\_ID=?", (de,))

 dme = tkinter.Label(rootDE, text="EMPLOYEE DELETED FROM DATABASE", fg="green")

 dme.place(x=20, y=100)

 conn.commit()

 else:

 error = tkinter.Label(rootDE, text="EMPLOYEE DOESN'T EXIST", fg="Red")

 error.place(x=20, y=100)

rootDE=None

def delo():

 global rootDE,d1

 rootDE=tkinter.Tk()

 rootDE.geometry("250x250")

 rootDE.title("EMPLOYEE DELETION")

 h1=tkinter.Label(rootDE,text="ENTER EMPLOYEE ID TO DELETE :")

 h1.place(x=20,y=10)

 d1=tkinter.Entry(rootDE)

 d1.place(x=20,y=40)

 B1=tkinter.Button(rootDE,text="DELETE",command=delling)

 B1.place(x=20,y=70)

 rootDE.mainloop()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

RooMT.py

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

import sqlite3

import tkinter

import tkinter.messagebox

conn=sqlite3.connect("MDBA.db")

P\_id=None

rootR=None

##ROOM BUTTON

def room\_button():

 global P\_id,r1,r2,room\_t,da,dd,rate,room\_no,r3,r4,r5,r6,conn

 conn = sqlite3.connect("MDBA.db")

 r1=P\_id.get()

 r2=room\_t.get(tkinter.ACTIVE)

 r3=room\_no.get(tkinter.ACTIVE)

 r4=rate.get()

 r5=da.get()

 r6=dd.get()

 conn.execute('INSERT INTO ROOM VALUES(?,?,?,?,?,?)',(r1,r3, r2, r4, r5, r6,))

 tkinter.messagebox.showinfo("MEDANTA DATABSE SYSTEM", "ROOM ALLOCATED")

 conn.commit()

 conn.close()

def update\_button():

 global P\_id,r1,r2,room\_t,da,dd,rate,room\_no,r3,r4,r5,r6,conn

 r1=P\_id.get()

 r2=room\_t.get(tkinter.ACTIVE)

 r3=room\_no.get(tkinter.ACTIVE)

 r4=rate.get()

 r5=da.get()

 r6=dd.get()

 p = list(conn.execute("Select \* from ROOM where PATIENT\_ID=?", (r1,)))

 if len(p) != 0:

 conn.execute('UPDATE ROOM SET ROOM\_NO=?,ROOM\_TYPE=?,RATE=?,DATE\_ADMITTED=?,DATE\_DISCHARGED=? where PATIENT\_ID=?',(r3, r2, r4, r5, r6,r1,))

 tkinter.messagebox.showinfo("MEDANTA DATABSE SYSTEM", "ROOM DETAILS UPDATED")

 conn.commit()

 else:

 tkinter.messagebox.showinfo("MEDANTA DATABSE SYSTEM", "PATIENT IS NOT ALLOCATED A ROOM")

##ROOT FOR DISPLAY ROOM INFO

rootRD=None

##EXIT FOR ROOM\_PAGE

def EXITT():

 global rootR

 rootR.destroy()

##FUNCTION FOR ROOM DISPLAY BUTTON

def ROOMD\_button():

 global r1,lr1,dis1,lr2,dis2,c1,ii,conn,c1,P\_iid

 conn = sqlite3.connect("MDBA.db")

 c1=conn.cursor()

 r1=P\_iid.get()

 p=list(c1.execute('select \* from ROOM where PATIENT\_ID=?',(r1,)))

 if (len(p)==0):

 tkinter.messagebox.showinfo("MEDANTA DATABASE SYSTEM","PATIENT NOT ALLOCATED ROOM")

 else:

 t=c1.execute('SELECT NAME,ROOM\_NO,ROOM\_TYPE FROM ROOM NATURAL JOIN PATIENT where PATIENT\_ID=?',(r1,));

 for ii in t:

 lr0=tkinter.Label(rootRD,text="PATIENT NAME",fg='blue')

 dis0=tkinter.Label(rootRD,text=ii[0])

 lr0.place(x=50,y=120)

 dis0.place(x=50,y=140)

 lr1=tkinter.Label(rootRD,text="ROOM NO",fg='blue')

 dis1=tkinter.Label(rootRD,text=ii[1])

 lr1.place(x=50,y=170)

 dis1.place(x=50,y=190)

 lr2=tkinter.Label(rootRD,text="ROOM TYPE",fg='blue')

 dis2=tkinter.Label(rootRD,text=ii[2])

 lr2.place(x=50,y=220)

 dis2.place(x=50,y=240)

def exittt():

 rootRD.destroy()

def roomDD():

 global rootRD,ra1,ss,P\_iid

 rootRD=tkinter.Tk()

 rootRD.geometry("280x280")

 rootRD.title("ROOM INFO")

 ra1=tkinter.Label(rootRD,text="ENTER PATIENT ID")

 ra1.place(x=20,y=20)

 P\_iid=tkinter.Entry(rootRD)

 ss=tkinter.Button(rootRD,text="SEARCH",command=ROOMD\_button)

 ra1.place(x=50, y=20)

 P\_iid.place(x=50, y=50)

 ss.place(x=70,y=80)

 e=tkinter.Button(rootRD,text="EXIT",command=exittt)

 e.place(x=150,y=80)

 rootRD.mainloop()

def exitt():

 rootR.destroy()

L=None

L1=None

def Room\_all():

 global rootR,r\_head,P\_id,id,room\_tl,L,i,room\_t,room\_nol,room\_no,L1,j,ratel,rate,da\_l,da ,dd\_l,dd,Submit,Update,cr

 rootR=tkinter.Tk()

 rootR.title("ROOM ALLOCATION")

 rootR.geometry("400x400")

 r\_head=tkinter.Label(rootR,text="ROOM ALLOCATION",font='Times 13 bold',fg="dark grey")

 r\_head.place(x=75,y=10)

 id=tkinter.Label(rootR,text="PATIENT ID")

 id.place(x=30,y=60)

 P\_id=tkinter.Entry(rootR)

 P\_id.place(x=100,y=60)

 room\_tl=tkinter.Label(rootR,text="ROOM TYPE")

 room\_tl.place(x=30, y=100)

 L=['SINGLE ROOM: Rs 4500','TWIN SHARING : Rs2500','TRIPLE SHARING: Rs2000']

 room\_t= tkinter.Listbox(rootR, width=22, height=3, selectmode='SINGLE', exportselection=0)

 for i in L:

 room\_t.insert(tkinter.END,i)

 room\_t.place(x=105,y=100)

 room\_nol=tkinter.Label(rootR,text="ROOM NUMBER")

 room\_nol.place(x=30,y=180)

 L1=['101','102-AA','102-BB','103','104-AA','104-BB','105','206-AAA','206-BBB','206-CCC','207','208-AAA','208-BBB','208-CCC','210','211','302','304-AA','304-BB']

 room\_no = tkinter.Listbox(rootR, width=8, height=1, selectmode='SINGLE', exportselection=0)

 for j in L1:

 room\_no.insert(tkinter.END,j)

 room\_no.place(x=135,y=180)

 ratel=tkinter.Label(rootR, text="ROOM CHARGES")

 ratel.place(x=30, y=220)

 rate=tkinter.Entry(rootR)

 rate.place(x=130, y=220)

 da\_l = tkinter.Label(rootR, text="DATE ADMITTED")

 da\_l.place(x=30,y=260)

 da=tkinter.Entry(rootR)

 da.place(x=140,y=260)

 dd\_l = tkinter.Label(rootR, text="DATE DISCHARGED(0)")

 dd\_l.place(x=30, y=300)

 dd =tkinter.Entry(rootR)

 dd.place(x=155, y=300)

 Submit=tkinter.Button(rootR,text="SUBMIT",command=room\_button)

 Submit.place(x=30,y=340)

 Update=tkinter.Button(rootR,text="UPDATE",command=update\_button)

 Update.place(x=130,y=340)

 cr=tkinter.Button(rootR,text='ROOM DETAILS',command=roomDD)

 cr.place(x=220,y=340)

 ee=tkinter.Button(rootR,text="EXIT",command=exitt)

 ee.place(x=330,y=340)

 rootR.mainloop()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

PATDELSU.py

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

import tkinter

import sqlite3

import tkinter.messagebox

conn=sqlite3.connect("MDBA.db")

#variables

rootU=None

rootD=None

rootS=None

head=None

inp\_s=None

searchB=None

#display/search button

def Search\_button():

 global inp\_s,entry,errorS,t,i,q,dis1,dis2,dis3,dis4,dis5,dis6,dis7,dis8,dis9,dis10

 global l1,l2,l3,l4,l5,l6,l7,l8,l9,l10

 c1=conn.cursor()

 inp\_s=entry.get()

 p=list(c1.execute('select \* from PATIENT where PATIENT\_ID=?',(inp\_s,)))

 if (len(p)==0):

 errorS=tkinter.Label(rootS,text="PATIENT RECORD NOT FOUND")

 errorS.pack()

 else:

 t=c1.execute('SELECT \* FROM PATIENT NATURAL JOIN CONTACT\_NO where PATIENT\_ID=?',(inp\_s,));

 for i in t:

 l1=tkinter.Label(rootS,text="PATIENT ID",fg='blue')

 dis1=tkinter.Label(rootS,text=i[0])

 l2=tkinter.Label(rootS,text="PATIENT NAME",fg='blue')

 dis2=tkinter.Label(rootS,text=i[1])

 l3=tkinter.Label(rootS,text="PATIENT SEX",fg='blue')

 dis3=tkinter.Label(rootS,text=i[2])

 l4=tkinter.Label(rootS,text="PATIENT BLOOD GROUP",fg='blue')

 dis4=tkinter.Label(rootS,text=i[3])

 l5=tkinter.Label(rootS,text="PATIENT DATE OF BIRTH",fg='blue')

 dis5=tkinter.Label(rootS,text=i[4])

 l6=tkinter.Label(rootS,text="PATIENT ADDRESS",fg='blue')

 dis6=tkinter.Label(rootS,text=i[5])

 l7=tkinter.Label(rootS,text="PATIENT DOCTOR/TEAM",fg='blue')

 dis7=tkinter.Label(rootS,text=i[6])

 l8=tkinter.Label(rootS,text="PATIENT EMAIL",fg='blue')

 dis8=tkinter.Label(rootS,text=i[7])

 l9=tkinter.Label(rootS,text="PATEINT CONTACT NO",fg='blue')

 dis9=tkinter.Label(rootS,text=i[8])

 l10=tkinter.Label(rootS,text="PATIENT ALTERNATE CONTACT",fg='blue')

 dis10=tkinter.Label(rootS,text=i[9])

 l1.pack()

 dis1.pack()

 l2.pack()

 dis2.pack()

 l3.pack()

 dis3.pack()

 l4.pack()

 dis4.pack()

 l5.pack()

 dis5.pack()

 l6.pack()

 dis6.pack()

 l7.pack()

 dis7.pack()

 l8.pack()

 dis8.pack()

 l9.pack()

 dis9.pack()

 l10.pack()

 dis10.pack()

 conn.commit()

def eXO():

 rootS.destroy()

##search window

def P\_display():

 global rootS,head,inp\_s,entry,searchB

 rootS=tkinter.Tk()

 rootS.title("SEARCH WINDOW")

 head=tkinter.Label(rootS,text="ENTER PATIENT ID TO SEARCH",fg="red")

 entry=tkinter.Entry(rootS)

 searchB=tkinter.Button(rootS,text='SEARCH',command=Search\_button)

 menubar= tkinter.Menu(rootS)

 filemenu = tkinter.Menu(menubar, tearoff=0)

 filemenu.add\_command(label="NEW", command=P\_display)

 filemenu.add\_separator()

 filemenu.add\_command(label="EXIT", command=eXO)

 menubar.add\_cascade(label="File", menu=filemenu)

 rootS.config(menu=menubar)

 head.pack()

 entry.pack()

 searchB.pack()

 rootS.mainloop()

inp\_d=None

entry1=None

errorD=None

disd1=None

#DELTE BUTTON

def Delete\_button():

 global inp\_d,entry1,errorD,disd1

 c1= conn.cursor()

 inp\_d = entry1.get()

 p=list(conn.execute("select \* from PATIENT where PATIENT\_ID=?", (inp\_d,)))

 if (len(p)==0):

 errorD = tkinter.Label(rootD, text="PATIENT RECORD NOT FOUND")

 errorD.pack()

 else:

 conn.execute('DELETE FROM PATIENT where PATIENT\_ID=?',(inp\_d,))

 disd1=tkinter.Label(rootD,text="PATIENT RECORD DELETED",fg='green')

 disd1.pack()

 conn.commit()

## DELETE SCREEN

def D\_display():

 global rootD,headD,inp\_d,entry1,DeleteB

 rootD=tkinter.Tk()

 rootD.title("DELETE WINDOW")

 headD=tkinter.Label(rootD,text="ENTER PATIENT ID TO DELETE",fg="blue")

 entry1=tkinter.Entry(rootD)

 DeleteB=tkinter.Button(rootD,text="DELETE",command=Delete\_button)

 headD.pack()

 entry1.pack()

 DeleteB.pack()

 rootD.mainloop()

##variables for update

pat\_ID=None

pat\_name=None

pat\_dob=None

pat\_address=None

pat\_sex=None

pat\_BG=None

pat\_email=None

pat\_contact=None

pat\_contactalt=None

pat\_CT=None

def up1():

 global u1, u2, u3, u4, u5, u6, u7, u8, u9, u10, ue1, conn

 conn.cursor()

 u1 = pat\_ID.get()

 u2 = pat\_name.get()

 u3 = pat\_sex.get()

 u4 = pat\_dob.get()

 u5 = pat\_BG.get()

 u6 = pat\_contact.get()

 u7 = pat\_contactalt.get()

 u8 = pat\_email.get()

 u9 = pat\_CT.get()

 u10 = pat\_address.get()

 conn = sqlite3.connect("MDBA.db")

 p = list(conn.execute("Select \* from PATIENT where PATIENT\_ID=?", (u1,)))

 if len(p) != 0:

 conn.execute('UPDATE PATIENT SET NAME=?,SEX=?,DOB=?,BLOOD\_GROUP=?,ADDRESS=?,CONSULT\_TEAM=?,EMAIL=? where PATIENT\_ID=?', ( u2, u3, u4, u5, u10, u9, u8,u1,))

 conn.execute('UPDATE CONTACT\_NO set CONTACTNO=?,ALT\_CONTACT=? WHERE PATIENT\_ID=?', ( u6, u7,u1,))

 tkinter.messagebox.showinfo("MEDANTA DATABSE SYSTEM", "DETAILS UPDATED INTO DATABASE")

 conn.commit()

 else:

 tkinter.messagebox.showinfo("MEDANTA DATABSE SYSTEM", "PATIENT IS NOT REGISTERED")

labelu=None

bu1=None

def EXITT():

 rootU.destroy()

##-----PATIENT UPDATE SCREEN -----##

def P\_UPDATE():

 global pat\_address, pat\_BG, pat\_contact, pat\_contactalt, pat\_CT, pat\_dob, pat\_email, pat\_ID, pat\_name, pat\_sex

 global rootU, regform, id, name, dob, sex, email, ct, addr, c1, c2, bg, SUBMIT, menubar, filemenu, p1f, p2f,HEAD

 rootU = tkinter.Tk()

 rootU.title("UPDATE WINDOW")

 menubar = tkinter.Menu(rootU)

 filemenu = tkinter.Menu(menubar, tearoff=0)

 filemenu.add\_command(label="NEW", command=P\_UPDATE)

 filemenu.add\_separator()

 filemenu.add\_command(label="EXIT", command=EXITT)

 rootU.config(menu=menubar)

 menubar.add\_cascade(label="File", menu=filemenu)

 HEAD=tkinter.Label(rootU,text="ENTER NEW DETAILS TO UPDATE",bg='black',fg='white')

 id = tkinter.Label(rootU, text="PATIENT ID")

 pat\_ID = tkinter.Entry(rootU)

 name = tkinter.Label(rootU, text="PATIENT NAME")

 pat\_name = tkinter.Entry(rootU)

 sex = tkinter.Label(rootU, text="SEX")

 pat\_sex = tkinter.Entry(rootU)

 dob = tkinter.Label(rootU, text="DOB (YYYY-MM-DD)")

 pat\_dob = tkinter.Entry(rootU)

 bg = tkinter.Label(rootU, text="BLOOD GROUP")

 pat\_BG = tkinter.Entry(rootU)

 c1 = tkinter.Label(rootU, text="CONTACT NUMBER")

 pat\_contact = tkinter.Entry(rootU)

 c2 = tkinter.Label(rootU, text="ALTERNATE CONTACT")

 pat\_contactalt = tkinter.Entry(rootU)

 email = tkinter.Label(rootU, text="EMAIL")

 pat\_email = tkinter.Entry(rootU)

 ct = tkinter.Label(rootU, text="CONSULTING TEAM / DOCTOR")

 pat\_CT = tkinter.Entry(rootU)

 addr = tkinter.Label(rootU, text="ADDRESS")

 pat\_address = tkinter.Entry(rootU)

 SUBMIT=tkinter.Button(rootU,text="SUBMIT",command=up1)

 HEAD.pack()

 id.pack()

 pat\_ID.pack()

 name.pack()

 pat\_name.pack()

 sex.pack()

 pat\_sex.pack()

 dob.pack()

 pat\_dob.pack()

 bg.pack()

 pat\_BG.pack()

 c1.pack()

 pat\_contact.pack()

 c2.pack()

 pat\_contactalt.pack()

 email.pack()

 pat\_email.pack()

 ct.pack()

 pat\_CT.pack()

 addr.pack()

 pat\_address.pack()

 SUBMIT.pack()

 rootU.mainloop()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

BILLING.py

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

import sqlite3

import tkinter

import tkinter.messagebox

conn=sqlite3.connect("MDBA.db")

#variables

rootB=None

def date\_up():

 global b1,b2

 b1 = P\_id.get()

 b2 = dd.get()

 conn.execute("UPDATE ROOM SET DATE\_DISCHARGED=? where PATIENT\_ID=?", (b2, b1,))

 conn.commit()

 tkinter.messagebox.showinfo("MEDANTA DATABASE SYSTEM", "DISCHARGE DATE UPDATED")

def up():

 global c1, b1, P\_id, b3, b4, b5, b6, dd, treat\_1, treat\_2, cost\_t, b7, b8, med, med\_q, price, u

 conn = sqlite3.connect("MDBA.db")

 c1 = conn.cursor()

 b1 = P\_id.get()

 b3 = treat\_1.get(tkinter.ACTIVE)

 b4 = treat\_2.get(tkinter.ACTIVE)

 b5 = cost\_t.get()

 b6 = med.get(tkinter.ACTIVE)

 b7 = med\_q.get(tkinter.ACTIVE)

 b8 = price.get()

 conn.execute("INSERT INTO TREATMENT VALUES(?,?,?,?)", (b1, b3, b4, b5,))

 conn.execute("INSERT INTO MEDICINE VALUES(?,?,?,?)", (b1, b6, b7, b8,))

 conn.commit()

 tkinter.messagebox.showinfo("MEDANTA DATABASE SYSTEM", "BILLING DATA SAVED")

def calci():

 global b1

 conn = sqlite3.connect("MDBA.db")

 u=conn.execute("Select sum(T\_COST+ (M\_COST\*M\_QTY) +(DATE\_DISCHARGED-DATE\_ADMITTED)\*RATE) FROM ROOM NATURAL JOIN TREATMENT natural JOIN MEDICINE where PATIENT\_ID=?",(b1,) )

 conn.commit()

 for ii in u:

 pp=tkinter.Label(rootB,text="TOTAL AMOUNT OUTSTANDING",fg='red',font='Arial 8 bold')

 pp.place(x="200", y='260')

 uu=tkinter.Label(rootB,text=ii[0])

 uu.place(x="230",y='290')

L1=None

L2=None

L3=None

L4=None

def exitt():

 rootB.destroy()

def BILLING():

 global rootB,L1,treat1,P\_id,dd,cost,med,med\_q,price,treat\_1,treat\_2,cost\_t,j,jj,jjj,jjjj,L2,L3,L4

 rootB=tkinter.Tk()

 rootB.geometry("450x350")

 rootB.title("BILLING SYSTEM")

 head=tkinter.Label(rootB,text="PATIENT BILL",font="Arial 16 bold italic",fg='grey')

 head.place(x=100,y=10)

 id = tkinter.Label(rootB, text="PATIENT ID")

 id.place(x=20, y=60)

 P\_id = tkinter.Entry(rootB)

 P\_id.place(x=100, y=60)

 dd\_l = tkinter.Label(rootB, text="DATE DISCHARGED")

 dd\_l.place(x=20, y=100)

 dd = tkinter.Entry(rootB)

 dd.place(x=135, y=100)

 ddp=tkinter.Button(rootB,text="UPDATE DISCHARGE DATE",command=date\_up)

 ddp.place(x=270,y=100)

 treat = tkinter.Label(rootB, text="SELECT TREATMENT")

 treat.place(x=20, y=140)

 L1 = ["CONSULATION","SURGERY","LAB TEST"]

 treat\_1= tkinter.Listbox(rootB, width=15, height=1, selectmode='SINGLE', exportselection=0)

 for j in L1:

 treat\_1.insert(tkinter.END, j)

 treat\_1.place(x=140,y=140)

 treat\_c = tkinter.Label(rootB, text="CODE")

 treat\_c.place(x=240, y=140)

 L2 = ["C\_1", "S\_1", "L\_1"]

 treat\_2 = tkinter.Listbox(rootB, width=6, height=1, selectmode='SINGLE', exportselection=0)

 for jj in L2:

 treat\_2.insert(tkinter.END, jj)

 treat\_2.place(x=280, y=140)

 costl= tkinter.Label(rootB, text="COST ?")

 costl.place(x=315, y=140)

 cost\_t = tkinter.Entry(rootB,width=5)

 cost\_t.place(x=350, y=140)

 med1 = tkinter.Label(rootB, text="SELECT MEDICINE")

 med1.place(x=20, y=180)

 L3 = ["NEURAL", "FANEKPLUS", "DISPRIN","DOLO+","BANDAGE","DIGENE"]

 med = tkinter.Listbox(rootB, width=15, height=1, selectmode='SINGLE', exportselection=0)

 for jjj in L3:

 med.insert(tkinter.END, jjj)

 med.place(x=140, y=180)

 med\_ql = tkinter.Label(rootB, text="QUANTITY")

 med\_ql.place(x=240, y=180)

 L4 = [1,2,3,4,5,6,7,8,9,10]

 med\_q = tkinter.Listbox(rootB, width=4, height=1, selectmode='SINGLE', exportselection=0)

 for jjjj in L4:

 med\_q.insert(tkinter.END, jjjj)

 med\_q.place(x=280, y=180)

 pricel = tkinter.Label(rootB, text="PRICE ?")

 pricel.place(x=315, y=180)

 price = tkinter.Entry(rootB, width=5)

 price.place(x=360, y=180)

 b1=tkinter.Button(rootB,text="GENERATE BILL",command=calci)

 b1.place(x="200",y="210")

 b2 = tkinter.Button(rootB, text="UPDATE DATA", command=up)

 b2.place(x="100", y="210")

 ee=tkinter.Button(rootB,text="EXIT",command=exitt)

 ee.place(x='310',y='210')

 rootB.mainloop()

**Block Diagram of the Model**

****

**B)Nonpreemptive Critical Section:**

b1 = 5

b2 = 5

b3 = 5

b4 = 5

           w1 (t) = 3 + 5 = 8

             W1 = 8 > p1 = 6

Therefore, T1 is not schedulable:

**Priority Ceiling:**



 w1(t) = 3 + 3 = 6
                                              W1 = 6 ≤ p1 = 6

             Continue using time demand analysis, to show that:

                                                    W2 = 18 ≤ p2 = 20
                                                    W3 = 52 ≤ p3 = 200
                                                    W4 = 53 ≤ p4 = 210

             Therefore, all tasks are schedulable



