**Code Entries (Listed by Group Member)**

# be3.py

# a program to introduce

# to maintain a list of Maldives hotel guest database

# by using the hotel management system program

# COMMENTS TO GIVE AUTHOR/VERSION HISTORY

Print “\t\tThis software allows the user to maintain”

Print “\t\tthe guest database to confirm, edit or delete the bookings”

**# TARIQ MAHMOOD 1ST GROUP MEMBER**

import sqlite3

def konnect():

con=sqlite3.connect("guest.db")

cur=con.cursor()

#print'created cursor'

cur.execute("CREATE TABLE IF NOT EXISTS hotelhotelguests(id integer PRIMARY KEY,fn text,ln text,In\_date text,Leave\_date text,bref integer)")

#print'created table'

con.commit()

con.close()

def addrec(fn,ln,In\_date,Leave\_date,bref):

konnect()

con=sqlite3.connect("guest.db")

cur=con.cursor()

cur.execute("INSERT INTO hotelhotelguests VALUES(NULL,?,?,?,?,?)",(fn,ln,In\_date,Leave\_date,bref))

print"just put this in :",(fn,ln,In\_date,Leave\_date,bref)

con.commit()

con.close()

def findit(fn='',ln='',In\_date='',Leave\_date='',bref=0):

con=sqlite3.connect('guest.db')

cur=con.cursor()

cur.execute("SELECT \* FROM hotelhotelguests WHERE FN=? OR LN=? OR In\_date=? OR Leave\_date=? OR bref=?",(fn,ln,In\_date,Leave\_date,bref))

rows=cur.fetchall()

con.close()

return rows

def update(mid,fn='',ln='',In\_date='',Leave\_date='',bref=0):

con=sqlite3.connect("guest.db")

cur=con.cursor()

cur.execute("UPDATE hotelhotelguests SET FN=?,LN=?,In\_date=?,Leave\_date=?,bref=? WHERE id=?",(fn,ln,In\_date,Leave\_date,bref,mid))

con.commit()

con.close()

def view():

con=sqlite3.connect("guest.db")

cur=con.cursor()

cur.execute("SELECT\*FROM hotelhotelguests")

row=cur.fetchall()

con.close()

return row

def dele(id):

konnect()

con=sqlite3.connect("guest.db")

cur=con.cursor()

cur.execute("DELETE FROM hotelhotelguests WHERE id=?",(id,))

con.commit()

con.close()

konnect()

from Tkinter import \*

import ttk

import be2 #backend

import sqlite3

#--------------------

rt=Tk()

rt.wm\_title('Hotel Management Sysytem')

**# JOINT EFFORT BY ALL MEMBERS IN THE GROUP**

#===============DEFINED FUNCTIONS=======================

def view\_command():

listbox.delete(0,END)

for row in be2.view():

listbox.insert(END,row)

def add\_command():

if(len(fn.get())!=0):

be2.addrec(fn.get(),ln.get(),in\_date.get(),Leave\_date.get(),bref.get())

listbox.delete(0,END)

listbox.insert(END,(fn.get(),ln.get(),in\_date.get(),Leave\_date.get(),bref.get()))

def dele\_command():

if(len(fn.get())!=0):

be2.dele(tt[0])

clearit\_command()

view\_command()

def gtr(event): #gtr=get the record

global tt # target tuple

entry\_id=listbox.curselection()[0]#gets the id number(this is at position 0) of the entry in listbox

tt=listbox.get(entry\_id)#get the record associated with entryt\_id

fn.set(tt[1])

ln.set(tt[2])

in\_date.set(tt[3])

Leave\_date.set(tt[4])

bref.set(tt[5])

text=''

def update\_command():

if(len(fn.get())!=0):#this checks to see if the first name box is not empty

be2.update(tt[0],fn.get(),ln.get(),in\_date.get(),Leave\_date.get(),bref.get())

view\_command()#refresh and update the list box

def findit\_command():

listbox.delete(0,END)#clear the listbox from the begining(0) to the end(END)

for row in be2.findit(fn.get(),ln.get(),in\_date.get(),Leave\_date.get(),bref.get()):

listbox.insert(END,row)#insert the values found by the "findit" function into the listbox

#starting at the begining.Since the listbox is empty the END is the begining

def clearEntry():#this function clears the entry boxes

fn.set(text) #text is an empty string ''

ln.set(text)

in\_date.set(text)

Leave\_date.set(text)

bref.set(text)

def dele\_command():

if(len(fn.get())!=0):#this checks to see if the first name box is not empty

be2.dele(tt[0])#tt[0] is the "entry id" of the record in the database

clearEntry()

view\_command()#refresh and update the list box

def correct(num):#this function restricts the type of keyboard entry to numbers only

#it will be used for the booking ref box

if num.isdigit():

return True

elif num is '': #an empty box is also acceptable

return True

else:

return False #no alphabet letters are allowed

def correct1(s1): #this function restricts the type of keyboard entry to alphabet only

if s1.isalpha():#it will be used for the first name and last name boxes

return True

else:

return False

**# BABATUNDE IFEOLUWA 2ND GROUP MEMBER**

#===============Labels for Vaiables===================

l1=Label(rt,text='First Name').grid(row=3,column=0)

l1a=Label(rt,text='Booking Ref.').grid(row=3,column=2)

l2=Label(rt,text='Last Name').grid(row=4,column=0)

l2a=Label(rt,text='in\_date').grid(row=5,column=0)

l2b=Label(rt,text='Leave\_date').grid(row=6,column=0)

#===========Data Entry Variables

fn=StringVar()

e1=Entry(rt,textvariable=fn)

e1.grid(row=3,column=1)

reg1=rt.register(correct1)

e1.config(validate="key",validatecommand=(reg1,'%P'))

Ln=StringVar()

e2=Entry(rt,textvariable=ln)

e2.grid(row=4,column=1)

reg1=rt.register(correct1)

e2.config(validate="key",validatecommand=(reg1,'%P'))

in\_date=StringVar()

e3=Entry(rt,textvariable=in\_date)

e3.grid(row=5,column=1)

Leave\_date=StringVar()

e4=Entry(rt,textvariable=Leave\_date)

e4.grid(row=6,column=1)

bref=StringVar ()

e1a=Entry(rt,textvariable=bref)

e1a.grid(row=3,column=3)

reg=rt.register(correct)

e1a.config(validate="key",validatecommand=(reg,'%P'))

Group 2 Member

The label allow the first name and the last name of a guest to be recorded.The entry box enters the hotel guest details via “fn”=First name entry

“ln”=Last name entry

Indate=entry box and

Outdate=entry box and the booking reference is automatically generated to secure accommodation or make necessary changes.

The text variable is equally declared as astring represented by “fn”The code when tested divides the attributes of the code intocolumn or rows.

**# OSMAN NEDUNCHEZHIAN 3RD GROUP MEMBER**

#===========LIST BOX==============================

yscrollbar=Scrollbar(rt)

yscrollbar.grid(row=6,column=4,rowspan=8)

listbox=Listbox(rt,height=6,width=75)

listbox.grid(row=7,column=0,columnspan=4)

listbox.configure(yscrollcommand=yscrollbar.set)

yscrollbar.configure(command=listbox.yview)

#===============================================

b1=Button(rt,text="List",width=10,command=view\_command)

b1.grid(row=0,column=0)

b2=Button(rt,text="Add Rec",width=10,command=add\_command)

b2.grid(row=0,column=1)

b3=Button(rt,text='Delete',width=10,command=dele\_command)

b3.grid(row=0,column=2)

b4=Button(rt,text="Update",width=10,command=update\_command)

b4.grid(row=0,column=3)

b5=Button(rt,text='Find',width=10,command=findit\_command)

b5.grid(row=0,column=4)

b6=Button(rt,text="Clear",width=10,command=clearEntry)

b6.grid(row=0,column=5)

b7=Button(rt,text='EXIT',bg='red',fg='white',width=10,command=rt.destroy)

b7.grid(row=0,column=6)

#===========================================================

listbox.bind('<<ListboxSelect>>',gtr)

rt.mainloop()