Customer / app.py

from tkinter import \*

import tkinter.messagebox

from tkinter import Tk

from db import Database

import os

from db import \*

import functools

db.reset\_all()

#Database file

db = Database(r'\\raspberrypi\share\jsj.db')

#Initialize

db.addcustomer("0", "0", "0")

# \*\*\*\* Functions \*\*\*\*

def addorange():

db.updateorange()

populate\_list()

populate\_totalp()

populate\_totalw()

def addapple():

db.updateapple()

populate\_list()

populate\_totalp()

populate\_totalw()

def addbanana():

db.updatebanana()

populate\_list()

populate\_totalp()

populate\_totalw()

def addtocart():

top = Toplevel()

top.title("Products")

top.geometry("300x300+730+280") #Window size

#Orange

button\_orange = Button(top, text="Orange", relief="raised", bd="3", height=2, width=4, command=addorange)

button\_orange.bind("<Button-1>")

button\_orange.place(x=15, y=15)

#Apple

button\_apple = Button(top, text="Apple", relief="raised", bd="3", height=2, width=4, command=addapple)

button\_apple.bind("<Button-1>")

button\_apple.place(x=85, y=15)

#Banana

button\_banana = Button(top, text="Banana", relief="raised", bd="3", height=2, width=4, command=addbanana)

button\_banana.bind("<Button-1>")

button\_banana.place(x=155, y=15)

#Close button

button\_close = Button(top, text="Close\nWindow", relief="raised", bd="3", height=2, width=5, command=top.destroy)

button\_close.bind("<Button-1>")

button\_close.place(x=125, y=230)

def newcust():

db.addcustomer("0", "0" , "0")

db.reset\_all()

populate\_list()

populate\_totalp()

populate\_totalw()

#Print

def checkout():

tkinter.messagebox.showinfo('JSJ Marketing by Group 10',

'Make sure to double check your items and thank you for shopping :)')

question = tkinter.messagebox.askquestion('Warning', 'Are sure you want to checkout?')

if question == 'yes':

os.system("sudo chmod a+w /dev/usb/lp0")

populate\_list()

populate\_totalp()

populate\_totalw()

pro = db.fetch\_orange()

proo = ''.join(str(e) for e in pro)

pra = db.fetch\_apple()

praa = ''.join(str(e) for e in pra)

prb = db.fetch\_banana()

prbb = ''.join(str(e) for e in prb)

prp = db.display\_price()

prpp = ''.join(str(e) for e in prp)

prw = db.display\_weight()

prww = ''.join(str(e) for e in prw)

proi = db.fetch\_orderid()

proid = ''.join(str(e) for e in proi)

#Receipt

os.system("sudo echo -e ' Item Qty Price(P) Weight(g) \n' > /dev/usb/lp0")

os.system("sudo echo -e '" +proo+ "' > /dev/usb/lp0")

os.system("sudo echo -e '" +praa+ "' > /dev/usb/lp0")

os.system("sudo echo -e '" +prbb+ "\n' > /dev/usb/lp0")

os.system("sudo echo -e 'Total Price(P): " +prpp+ "' > /dev/usb/lp0")

os.system("sudo echo -e 'Total Weight(g): " +prww+ "\n\n' > /dev/usb/lp0")

os.system("sudo echo -e 'Order ID: " +proid+ "\n' > /dev/usb/lp0")

os.system("sudo echo -e 'Thank you for Shopping! \nJSJ Marketing \n\n\n' > /dev/usb/lp0")

db.upinvent()

if question == 'no':

print('Enjoy shopping')

def populate\_list():

parts\_list.delete(0, END)

for row in db.fetch():

parts\_list.insert(END, row)

def populate\_totalp():

displaytotalp.delete(0, END)

for roww in db.display\_price():

displaytotalp.insert(END, roww)

def populate\_totalw():

displaytotalw.delete(0, END)

for rowww in db.display\_weight():

displaytotalw.insert(END, rowww)

def select\_item(event):

try:

global selected\_item

index = parts\_list.curselection()[0]

selected\_item = parts\_list.get(index)

print(selected\_item)

except IndexError:

pass

def remove\_item():

db.remove(selected\_item[0])

populate\_list()

populate\_totalp()

populate\_totalw()

#Reset button

def resetbox():

tkinter.messagebox.showinfo('JSJ Marketing by Group 10',

'Make sure to double check your items')

question = tkinter.messagebox.askquestion('Warning', 'Are sure you want to reset all items?')

if question == 'yes':

db.reset\_all()

db.resetbox1()

populate\_list()

populate\_totalp()

populate\_totalw()

if question == 'no':

print('Enjoy shopping')

#Main Window

root = Tk()

root.title('JSJ Marketing by Group 10')

#Logo

logoPhoto = PhotoImage(file="ui/logoz.png")

logophotolabel = Label(root, image=logoPhoto)

logophotolabel.place(x=15, y=15, anchor=NW)

# \*\*\*\* Picture button. ADD TO CART. DELETE ITEM. PRINT ALL ITEM \*\*\*\*

photoadd = PhotoImage(file="ui/addsz.png")

button\_1 = Button(root, image=photoadd, relief="raised", bd="3", command=addtocart)

button\_1.bind("<Button-1>", addtocart)

button\_1.place(x=40, y=550)

photodel = PhotoImage(file="ui/delete.png")

button\_2 = Button(root, image=photodel, relief="raised", bd="3", command=remove\_item)

button\_2.bind("<Button-1>")

button\_2.place(x=160, y=550)

photoprint = PhotoImage(file="ui/print.png")

button\_3 = Button(root, image=photoprint, relief="raised", bd="3", command=checkout)

button\_3.bind("<Button-1>", checkout)

button\_3.place(x=420, y=550)

photores = PhotoImage(file="ui/reset.png")

button\_4 = Button(root, image=photores, relief="raised", bd="3", command=resetbox)

button\_4.bind("<Button-1>")

button\_4.place(x=310, y=550)

#New Customer Button

button\_newcust = Button(root, text="New Customer", relief="raised", bd="3", height=2, width=13, font = ('Roboto',10), command=newcust)

button\_newcust.bind("<Button-1>")

button\_newcust.place(x=40, y=400)

"""TEXTS"""

item\_text = StringVar()

itemlabel = Label(root, text='ITEM')

itemlabel.place(x=120, y=120)

quantity\_text = StringVar()

quantitylabel = Label(root, text='QTY')

quantitylabel.place(x=200, y=120)

price\_text = StringVar()

pricelabel = Label(root, text='PRICE(₱)')

pricelabel.place(x=230, y=120)

weight\_text = StringVar()

weightlabel = Label(root, text='WEIGHT(g)')

weightlabel.place(x=310, y=120)

#Item List (Listbox)

parts\_list = Listbox(root, relief="raised", height=5, width=20, border=0, font = ('Roboto',30))

parts\_list.grid(padx=40, pady=138, columnspan=3, rowspan=6) #columnspan=3, rowspan=6, pady=10, padx=20)

parts\_list.bind('<<ListboxSelect>>', select\_item)

#Total Price (Listbox)

displaytotalp = Listbox(root, relief="raised", height=1, width=10, border=0, font = ('Roboto',14))

displaytotalp.place(x=365, y=420)

displaytotalp.bind('<<ListboxSelect>>', select\_item)

totalprice = StringVar()

totalpricelabel = Label(root, text='Total Price(₱)', font = ('Roboto',13))

totalpricelabel.place(x=225, y=420)

totalprice\_entry = Entry(root, textvariable=totalprice)

#Total Weight (Listbox)

displaytotalw = Listbox(root, relief="raised", height=1, width=10, border=0, font = ('Roboto',14))

displaytotalw.place(x=365, y=441)

displaytotalw.bind('<<ListboxSelect>>', select\_item)

totalweight = StringVar()

totalweightlabel = Label(root, text='Total Weight(g)', font = ('Roboto',13))

totalweightlabel.place(x=225, y=442)

totalweight\_entry = Entry(root, textvariable=totalweight)

# Create scrollbar

scrollbar = Scrollbar(root, width=20, border=0)

scrollbar.place(x=509, y=280, anchor=W)

# Set scroll to listbox

parts\_list.configure(yscrollcommand=scrollbar.set)

scrollbar.configure(command=parts\_list.yview)

# Bind select

parts\_list.bind('<<ListboxSelect>>', select\_item)

populate\_list()

populate\_totalp()

populate\_totalw()

root.geometry('560x680+600+3') #Window size

root.mainloop()

Customer / db.py

import sqlite3

class Database:

def \_\_init\_\_(self, db):

self.conn = sqlite3.connect(db)

self.cur = self.conn.cursor()

self.cur.execute(

"CREATE TABLE IF NOT EXISTS parts (id INTEGER PRIMARY KEY, item text, quantity INTEGER, price INTEGER, weight INTEGER)")

self.conn.commit()

def fetch(self):

self.cur.execute("SELECT id, item, quantity, price, weight FROM parts WHERE quantity > 0")

rows = self.cur.fetchall()

return rows

def addcustomer(self,orange, apple, banana):

self.cur.execute("INSERT INTO orders VALUES (NULL, ?, ?, ?)",

(orange, apple, banana))

self.conn.commit()

def remove(self, id1):

self.cur.execute("UPDATE parts SET quantity = 0, price = 0, weight = 0 WHERE id = ?", (id1,))

self.conn.commit()

if id1 == 1:

self.cur.execute("UPDATE orders SET orange = 0 WHERE orderid = (SELECT orderid FROM orders ORDER BY orderid DESC LIMIT 1)")

self.conn.commit()

elif id1 == 2:

self.cur.execute("UPDATE orders SET apple = 0 WHERE orderid = (SELECT orderid FROM orders ORDER BY orderid DESC LIMIT 1)")

self.conn.commit()

elif id1 == 3:

self.cur.execute("UPDATE orders SET banana = 0 WHERE orderid = (SELECT orderid FROM orders ORDER BY orderid DESC LIMIT 1)")

self.conn.commit()

def reset\_all(self):

self.cur.execute("UPDATE parts SET quantity = 0, price = 0, weight = 0")

self.conn.commit()

def updateorange(self):

self.cur.execute("UPDATE orders SET orange = orange + 1 WHERE orderid =(SELECT orderid from orders ORDER BY orderid DESC LIMIT 1)")

self.cur.execute("UPDATE parts SET quantity = (SELECT orange from orders WHERE orderid = (SELECT orderid from orders ORDER BY orderid DESC LIMIT 1)),\

price = 10\*(SELECT orange from orders WHERE orderid = (SELECT orderid from orders ORDER BY orderid DESC LIMIT 1)),\

weight = 135\*(SELECT orange from orders WHERE orderid = (SELECT orderid from orders ORDER BY orderid DESC LIMIT 1))\

WHERE id = 1")

self.conn.commit()

def updateapple(self):

self.cur.execute("UPDATE orders SET apple = apple + 1 WHERE orderid =(SELECT orderid from orders ORDER BY orderid DESC LIMIT 1)")

self.cur.execute("UPDATE parts SET quantity = (SELECT apple from orders WHERE orderid = (SELECT orderid from orders ORDER BY orderid DESC LIMIT 1)),\

price = 10\*(SELECT apple from orders WHERE orderid = (SELECT orderid from orders ORDER BY orderid DESC LIMIT 1)),\

weight = 145\*(SELECT apple from orders WHERE orderid = (SELECT orderid from orders ORDER BY orderid DESC LIMIT 1))\

WHERE id = 2")

self.conn.commit()

def updatebanana(self):

self.cur.execute("UPDATE orders SET banana = banana + 1 WHERE orderid =(SELECT orderid from orders ORDER BY orderid DESC LIMIT 1)")

self.cur.execute("UPDATE parts SET quantity = (SELECT banana from orders WHERE orderid = (SELECT orderid from orders ORDER BY orderid DESC LIMIT 1)),\

price = 8\*(SELECT banana from orders WHERE orderid = (SELECT orderid from orders ORDER BY orderid DESC LIMIT 1)),\

weight = 120\*(SELECT banana from orders WHERE orderid = (SELECT orderid from orders ORDER BY orderid DESC LIMIT 1))\

WHERE id = 3")

self.conn.commit()

def display\_price(self):

self.cur.execute("SELECT SUM(price) FROM parts")

tprice= self.cur.fetchall()

return tprice

def display\_weight(self):

self.cur.execute("SELECT SUM(weight) FROM parts")

tweight= self.cur.fetchall()

return tweight

def fetch\_orange(self):

self.cur.execute("SELECT id, item, quantity, price, weight FROM parts where id = 1 AND quantity > 0")

forange= self.cur.fetchall()

return forange

def fetch\_apple(self):

self.cur.execute("SELECT id, item, quantity, price, weight FROM parts where id = 2 AND quantity > 0")

fapple= self.cur.fetchall()

return fapple

def fetch\_banana(self):

self.cur.execute("SELECT id, item, quantity, price, weight FROM parts where id = 3 AND quantity > 0")

fbanana= self.cur.fetchall()

return fbanana

def fetch\_orderid(self):

self.cur.execute("SELECT orderid FROM orders ORDER BY orderid DESC LIMIT 1")

forderid = self.cur.fetchall()

return forderid

def upinvent(self):

self.cur.execute("UPDATE inventory SET quantity = quantity - (SELECT orange FROM orders ORDER BY orderid DESC LIMIT 1) WHERE id = 1")

self.conn.commit()

self.cur.execute("UPDATE inventory SET quantity = quantity - (SELECT apple FROM orders ORDER BY orderid DESC LIMIT 1) WHERE id = 2")

self.conn.commit()

self.cur.execute("UPDATE inventory SET quantity = quantity - (SELECT banana FROM orders ORDER BY orderid DESC LIMIT 1) WHERE id = 3")

self.conn.commit()

def resetbox1(self):

self.cur.execute("UPDATE orders SET orange = 0, apple = 0, banana = 0 WHERE orderid = (SELECT orderid FROM orders ORDER BY orderid DESC LIMIT 1)")

self.conn.commit()

def \_\_del\_\_(self):

self.conn.close()

db = Database(r'\\raspberrypi\share\jsj.db')