Fatih Güven
191013017

```
from tkinter import *
from tkinter import ttk
import sqlite3
from models import *
from events import *
import threading, time

# Veritabanı bağlantısını kur
db = sqlite3.connect("mailler.db")

# tabloları oluştur
initTables(db)

main = None

class mainForm():
root = None

currentmail = None

def __init__(self):
# pencereyi oluştur
self.root = Tk()
self.root.title("Toplu Mail Yazılımı")
self.root.geometry("640x420")
self.root.resizable(False, False)

Button(self.root, text="Yeni Hesap Ekle",
command=self.addnewaccoount).place(x=500,y=5)
# gönderme işleminin yapılacağı mail hesabı
self.currentmail = ttk.Combobox(self.root, state="readonly", width=30)
self.currentmail.place(x=230, y=11)

# alıcılara mail ekle
self.addreceiverbtn = Button(self.root, text="+", width=4, command=self.addreceiver)
self.addreceiverbtn.place(y=5,x=5)

# alıcı listesini yenile
self.addreceiverbtn = Button(self.root, text="↻", width=4,
command=self.updateCurrentMails)
self.addreceiverbtn.place(y=5,x=73)

# seçili alıcıyı sil
self.addreceiverbtn = Button(self.root, text="-", width=4, command=self.deletereceiver)
self.addreceiverbtn.place(y=5,x=141)

# mail listesi
```

```
self.maillist = Listbox(self.root, width=25, height=21)
self.maillist.place(y=40,x=0)

# gönderilecek mail içerikleri
Label(self.root, text="Başlık").place(x=220, y=40)
self.title = Entry(self.root, width=44)
self.title.place(x=270, y=40)
Label(self.root, text="Mesaj").place(x=220, y=70)
self.message = Text(self.root, width=50, height=16)
self.message.place(x=220, y=90)

# mail gönder
self.sendmailbtn = Button(self.root, text="Gönder", width=10, command=self.sendmails)
self.sendmailbtn.place(x=515,y=380)

self.sendstatus = Label(self.root, text="")
self.sendstatus.place(x=220,y=380)

# başlangıç güncellemeleri
self.updateCurrentMails()
self.currentmail.current(0)
self.updateReceiverList()

self.root.mainloop()

def sendmails(self):
global db
c = db.cursor()
c.execute("SELECT oid, name, mail FROM `mailaddress` ORDER BY RANDOM()")
maillist = c.fetchall()
db.commit()
mlist = []
for receiver in maillist:
mlist.append({
"oid": receiver[0],
"name": receiver[1],
"mail": receiver[2]
})

title = self.title.get()
message = self.message.get("1.0",END)
c = db.cursor()
c.execute("SELECT * FROM `smtp` WHERE mail=:mail", {'mail':self.currentmail.get()})
smtpsettings = c.fetchone()
db.commit()

key = b'Rd_SYbKvCa33Bk6ceRrTnZJpYYYqVU5kJs_OvYV7ZEM='
settings = {
"server": smtpsettings[3],
"mail": smtpsettings[1],
"passwd": decrypt(smtpsettings[2], key.decode()).decode(),
```

```
"port": int(smtpsettings[4]),
}

self.sendstatus['text'] = "Giriş yapılıyor"

x = threading.Thread(target=self.sendmails_thread, args=(title, message, settings, mlist,))
x.start()

def sendmails_thread(self, title, message, settings, maillist):
self.sendstatus['foreground'] = "dark green"
for s in sendMail(title, message, settings, maillist):
self.sendstatus['text'] = s['message']
if s['status'] == False:
self.sendstatus['foreground'] = "dark red"
return

self.sendstatus['text'] = "Başarıyla tamamlandı..."


def deletereceiver(self):
global db

indis = self.maillist.curselection()

if indis:
selected = self.maillist.get(indis)
c = db.cursor()

c.execute("SELECT oid FROM `mailaddress` WHERE mail=:mail",{
"mail": selected
})

oid = c.fetchone()[0]
if oid:
c.execute("DELETE FROM `mailaddress` WHERE oid=:oid", {
"oid": oid
})

db.commit()

self.updateReceiverList()

def addreceiver(self):
formNewReceiver(self)

def updateReceiverList(self):
global db
c = db.cursor()
c.execute("SELECT mail FROM `mailaddress` ORDER BY mail DESC")
maillist = c.fetchall()
db.commit()
```

```
maillist = list(map(lambda x: x[0], maillist))

self.maillist.delete(0, 'end')
for mail in maillist:
self.maillist.insert(0, mail)

def updateCurrentMails(self):
global db
c = db.cursor()
c.execute("SELECT mail FROM `smtp` ORDER BY mail DESC")
maillist = c.fetchall()
db.commit()

maillist = list(map(lambda x: x[0], maillist))

self.currentmail['values'] = maillist

def addnewaccoount(self):
formAddSmtp(self)


class formAddSmtp():

def __init__(self, main):
self.main = main

self.root = Tk()
self.root.title("Yeni Mail Hesabı Ekle")
self.root.geometry("370x280")

self.root.resizable(False, False)

Label(self.root, text="Yeni Mail Hesabı Ekle",
justify=CENTER).grid(row=0,column=0,columnspan=2, padx=5, pady=5)
Label(self.root, text="Gönderici İsmi", justify=LEFT).grid(row=1,column=0, sticky='w',
padx=5)
Label(self.root, text="Mail Adresi", justify=LEFT).grid(row=2,column=0, sticky='w',
padx=5)
Label(self.root, text="Parola", justify=LEFT).grid(row=3,column=0, sticky='w', padx=5)
Label(self.root, text="Sunucu", justify=LEFT).grid(row=4,column=0, sticky='w', padx=5)
Label(self.root, text="Port", justify=LEFT).grid(row=5,column=0, sticky='w', padx=5)

self.sendername = Entry(self.root, width=30)
self.mail = Entry(self.root, width=30)
self.passwd = Entry(self.root, show="*", width=30)
self.server = Entry(self.root, width=30)
self.port = ttk.Combobox(self.root, state="readonly", width=29)

self.sendername.grid(row=1,column=1, sticky='w',pady=5, padx=5)
self.mail.grid(row=2,column=1, sticky='w',pady=5, padx=5)
```

```
self.passwd.grid(row=3,column=1, sticky='w',pady=5, padx=5)
self.server.grid(row=4,column=1, sticky='w',pady=5, padx=5)
self.port.grid(row=5,column=1, sticky='w',pady=5, padx=5)
self.port['values'] = ('465', '587')
self.port.current(0)

Button(self.root, text="Test Et", command=self.maketest).grid(row=7, column=0,
columnspan=2, pady=5, padx=5, sticky='nesw')
Button(self.root, text="Iptal", command=lambda: self.root.destroy()).grid(row=8,
column=0, columnspan=2, pady=5, padx=5, sticky='nesw')

self.root.mainloop()

def maketest(self):
settings = {
"server": self.server.get(),
"mail": self.mail.get(),
"passwd": self.passwd.get(),
"port": int(self.port.get()),
}
receiverlist = [
{
"name": self.sendername.get(),
"mail": self.mail.get(),
}
]
title = "Test Mail"
message = "Bu bir sınama mailidir. Lütfen bunu dikkate almayın!"

state = False
for s in sendMail(title, message, settings, receiverlist):
state = s['status']

if state == True:
self.sendername.config(state='disabled')
self.mail.config(state='disabled')
self.passwd.config(state='disabled')
self.server.config(state='disabled')
self.port.config(state='disabled')
Button(self.root, text="Kaydet", command=self.savesmtp).grid(row=7, column=0,
columnspan=2, pady=5, padx=5, sticky='nesw')

def savesmtp(self):
global db

key = b'Rd_SYbKvCa33Bk6ceRrTnZJpYYYqVU5kJs_OvYV7ZEM='
passwd = self.passwd.get()
passwd = encrypt(passwd.encode(), key.decode())

settings = {
"sendername": self.sendername.get(),
```

```
"server": self.server.get(),
"mail": self.mail.get(),
"passwd": passwd,
"port": int(self.port.get()),
}

c = db.cursor()
c.execute("INSERT INTO `smtp` VALUES (:sendername, :mail, :passwd, :server, :port)",
settings)
db.commit()

self.main.updateCurrentMails()

self.root.destroy()

class formNewReceiver():
def __init__(self, main):
self.main = main

self.root = Tk()
self.root.title("Yeni Kişi Ekle")
self.root.geometry("350x230")

self.root.resizable(False, False)

Label(self.root, text="Yeni Kişi Ekle",
justify=CENTER).grid(row=0,column=0,columnspan=2, padx=5, pady=5)
Label(self.root, text="Alıcı İsmi", justify=LEFT).grid(row=1,column=0, sticky='w', padx=5)
Label(self.root, text="Mail Adresi", justify=LEFT).grid(row=2,column=0, sticky='w',
padx=5)

self.name = Entry(self.root, width=30)
self.mail = Entry(self.root, width=30)

self.name.grid(row=1,column=1, sticky='w',pady=5, padx=5)
self.mail.grid(row=2,column=1, sticky='w',pady=5, padx=5)

Button(self.root, text="Kaydet", command=self.savemail).grid(row=4, column=0,
columnspan=2, pady=5, padx=5, sticky='nesw')
Button(self.root, text="Iptal", command=lambda: self.root.destroy()).grid(row=5,
column=0, columnspan=2, pady=5, padx=5, sticky='nesw')

self.statustext = Label(self.root, text="", justify=CENTER, foreground="dark red")
self.statustext.grid(row=3,column=0, padx=5, columnspan=2)

self.root.mainloop()

def savemail(self):
global db
email = self.mail.get()
name = self.name.get()
```

```
if checkmail(email):
try:

if len(name) > 3:
c = db.cursor()

c.execute("SELECT count(*) FROM `mailaddress` WHERE mail=:mail",{
"mail": email
})

mailcount = c.fetchone()[0]

if mailcount != 0:
self.statustext['text'] = "Bu mail daha önce eklenmiş!"
self.statustext['foreground'] = "dark red"

else:
c.execute("INSERT INTO `mailaddress` VALUES (:name, :mail, 0)",{
"name": name,
"mail": email
})

self.statustext['text'] = "Yeni kişi eklendi"
self.statustext['foreground'] = "dark green"
self.name.delete(0, 'end')
self.mail.delete(0, 'end')
self.main.updateReceiverList()

db.commit()

else:
self.statustext['text'] = "Kişinin ismini kontrol ediniz!"
self.statustext['foreground'] = "dark red"
except:
self.statustext['text'] = "Kişi kaydedilemedi!"
self.statustext['foreground'] = "dark red"
else:
self.statustext['text'] = "Girdiğiniz mail adresi geçersiz!"
self.statustext['foreground'] = "dark red"


main = mainForm()

db.close()
```