Solution:

```
from tkinter import *
from tkinter import messagebox
import pickle


class listObject():

    def __init__(self, name1, password1, balance1):
        self._name = name1
        self._password = password1
        self._balance = balance1

    def setBalance(self, balance1):
        self._balance=balance1

    def setName(self, name1):
        self._name=name1

    def addBalance(self, money):
        self._balance=self._balance+money

    def subBalance(self, money):
        self._balance=self._balance-money


    def setpassword(self, password1):
        self._password=password1

    def getBalance(self):
        return self._balance

    def getName(self):
        return self._name

    def getPassword(self):
        return self._password


classList = {}

window = Tk()



try:
    with open('database.pickle', 'rb') as fileIn:
        while True:
            try:
                classList = pickle.load(fileIn)
            except EOFError:
                break
    fileIn.close()
```

```
except OSError as e:
    pass




pickleFileOut = open('database.pickle', 'wb')


def saveDataOnExit():
    pickle.dump(classList, pickleFileOut)
    pickleFileOut.close()
    window.destroy()


window.protocol('WM_DELETE_WINDOW', saveDataOnExit)  # root is your root
window




def mainWindow():
    clearWindow()
    window.title("Welcome to ATM machine")
    window.geometry('350x100')
    btn1 = Button(window, text = "Sign in", command = signIn)
    btn2 = Button(window, text = "Add account", command = addAccount)
    btn3 = Button(window, text = "Delete account", command = deleteAccount)
    btn4 = Button(window, text = "Show accounts", command = showAccount)
    btn5 = Button(window, text = "Delete all accounts", command = destroyAll)
    btn1.grid(column=0, row=0)
    btn2.grid(column=1, row=0)
    btn3.grid(column=0, row=1)
    btn4.grid(column=1, row=1)
    btn5.grid(column=3, row =1)


def deleteAccount():
    print("Working?")
    clearWindow()
    lbl = Label(window, text = "Enter the username you want to delete:")
    txt = Entry(window, width=20)
    btn = Button(window, text = "Enter", command = lambda:
deleted(txt.get()))
    btn2 = Button(window, text = "Main Menu", command = mainWindow)
    lbl.grid(column=0, row = 0)
    txt.grid(column=1, row = 0)
    btn.grid(column=0, row = 1)
    btn2.grid(column=1, row = 1)


def deleted(name):
    for k in classList.keys():
        if(name == classList[k].getName()):
```

```
            del classList[k]
            mainWindow()
            return
    messagebox.showinfo("Error", "No account exists")
    mainWindow()



def destroyAll():
    classList.clear()


def showAccount():
    clearWindow()
    #for k in list.items():
    textString=""
    for k in classList.keys():
        textString += classList[k].getName()+ "-" +
str(classList[k].getBalance()) + "-" + classList[k].getPassword() + "\n"

    label = Label(window, text = textString)
    label2 = Label(window, text = "NAME - BALANCE - PASSWORD: ")
    label2.grid(column = 0, row = 0)
    label.grid(column = 0, row = 1)
    btn = Button(window, text = "Main menu", command = mainWindow)
    btn.grid(column = 1, row = 2)




def signIn():
    clearWindow()
    window.title("Welcome to ATM machine")
    window.geometry('350x200')
    lbl = Label(window, text="Username")
    lb2 = Label(window, text="Password")
    lbl.grid(column=0, row=0)
    lb2.grid(column=0, row=1)
    txt = Entry(window, width=20)
    txt2 = Entry(window, show='*', width=20)
    txt.grid(column=1, row=0)
    txt2.grid(column=1, row=1)
    button = Button(window, text = "Enter", command =
lambda:signedIn(txt.get(), txt2.get()))
    button.grid(column=2, row=1)
    btn = Button(window, text="Main menu", command=mainWindow)
    btn.grid(column=1, row=2)


def signedIn(username, password):
    if(len(classList)==0):
        messagebox.showinfo("Error", "No accounts available")
        mainWindow()
        return
    else:
        for k in classList.keys():
            if(classList[k].getName() == username):
```

```
                if(classList[k].getPassword() != password):
                    signIn()
                    messagebox.showinfo("Error", "Wrong password, please try
again")
                    return

                else:
                    clearWindow()
                    window.title("Welcome to ATM machine")
                    window.geometry('350x200')
                    bt1 = Button(window, text="Withdraw", command=
lambda:withdraw(username))
                    bt2 = Button(window, text="Balance inquiry",
command=lambda:balance(username))
                    bt3 = Button(window, text="Deposit",
command=lambda:deposit(username))
                    bt1.grid(column=0, row=0)
                    bt2.grid(column=0, row=1)
                    bt3.grid(column=0, row=2)
                    btn = Button(window, text="Main menu",
command=mainWindow)
                    btn.grid(column=1, row=2)
                    return
    signIn()
    messagebox.showinfo("Error", "No username exists, please try again")


def withdraw(username):
    clearWindow()
    label = Label(window, text = "Amount:")
    txt = Entry(window, width= 20)
    label.grid(column=0, row=0)
    txt.grid(column=1, row=0)
    bt = Button(window, text = "Enter", command =
lambda:withdrawMoney(username,txt))
    bt.grid(column = 1, row = 1)


def withdrawMoney(username, money):
    try:
        if(float(money.get()) == ''):
            withdraw(username)
        else:
            classList[username].subBalance(float(money.get()))
            mainWindow()
    except ValueError:
        withdraw(username)




def balance(username):
    clearWindow()
    label = Label(window, text="Amount:" +
str(classList[username].getBalance()))
    #txt = Entry(window, width=20)
```

```
    label.grid(column=0, row=0)
    #txt.grid(column=1, row=0)
    bt = Button(window, text="Main Menu", command=mainWindow)
    bt.grid(column=1, row=1)

def depositMoney(username, money):
    try:
        if (float(money.get()) == ''):
            deposit(username)
        else:
            classList[username].addBalance(float(money.get()))
            mainWindow()
    except ValueError:
        deposit(username)

def deposit(username):
    clearWindow()
    label = Label(window, text="Amount:")
    txt = Entry(window, width=20)
    label.grid(column=0, row=0)
    txt.grid(column=1, row=0)
    bt = Button(window, text="Enter", command=lambda:depositMoney(username,
txt))
    bt.grid(column=1, row=1)


def addAccount():
    clearWindow()
    window.title("Welcome to ATM machine")
    window.geometry('350x200')
    lbl = Label(window, text="Please enter a username")
    lb2 = Label(window, text="Please enter a Password")
    lb3 = Label(window, text="Please reenter password")
    lb4 = Label(window, text="Please enter initial deposit")
    lbl.grid(column=0, row=0)
    lb2.grid(column=0, row=1)
    lb3.grid(column=0, row=2)
    lb4.grid(column=0, row=3)
    txt = Entry(window, width=20)
    txt2 = Entry(window, show='*', width=20)
    txt3 = Entry(window, show='*', width=20)
    txt4 = Entry(window, width=20)
    txt.grid(column=1, row=0)
    txt2.grid(column=1, row=1)
    txt3.grid(column=1, row=2)
    txt4.grid(column=1, row=3)

    btn = Button(window, text = "Enter", command =
lambda:addedAccount(txt,txt2,txt3, txt4))
    btn.grid(column=1, row=4)
    btn1 = Button(window, text = "Main Menu", command = mainWindow)
    btn1.grid(column=2, row =4)

def addedAccount(name, password, password1, amount):

    if(password.get() != password1.get()):
        messagebox.showinfo('Error', 'Passwords do not match, try again!')
```

```
        addAccount()
        return
    if(name.get()=='' or password.get()=='' or amount.get()==''):
        messagebox.showinfo('Error', 'Please try again! There is a blank
entry.')
        addAccount()
        return
    for key, value in classList.items():
        print(key)
        if(key == name.get()):
            messagebox.showinfo("Error", "Account already exists, please try
again!")
            addAccount()
            return
    else:
        try:
            p = listObject(name.get(),password.get(),float(amount.get()))
            classList[name.get()]=p
        #pickle.dump(classList, file)
            mainWindow()
        except ValueError:
            messagebox.showinfo("Error", "Sorry please only put numerical
amount in initial deposit")
            addAccount()



def clearWindow():
    list=window.grid_slaves()
    for l in list:
        l.destroy()



mainWindow()



window.mainloop()
```