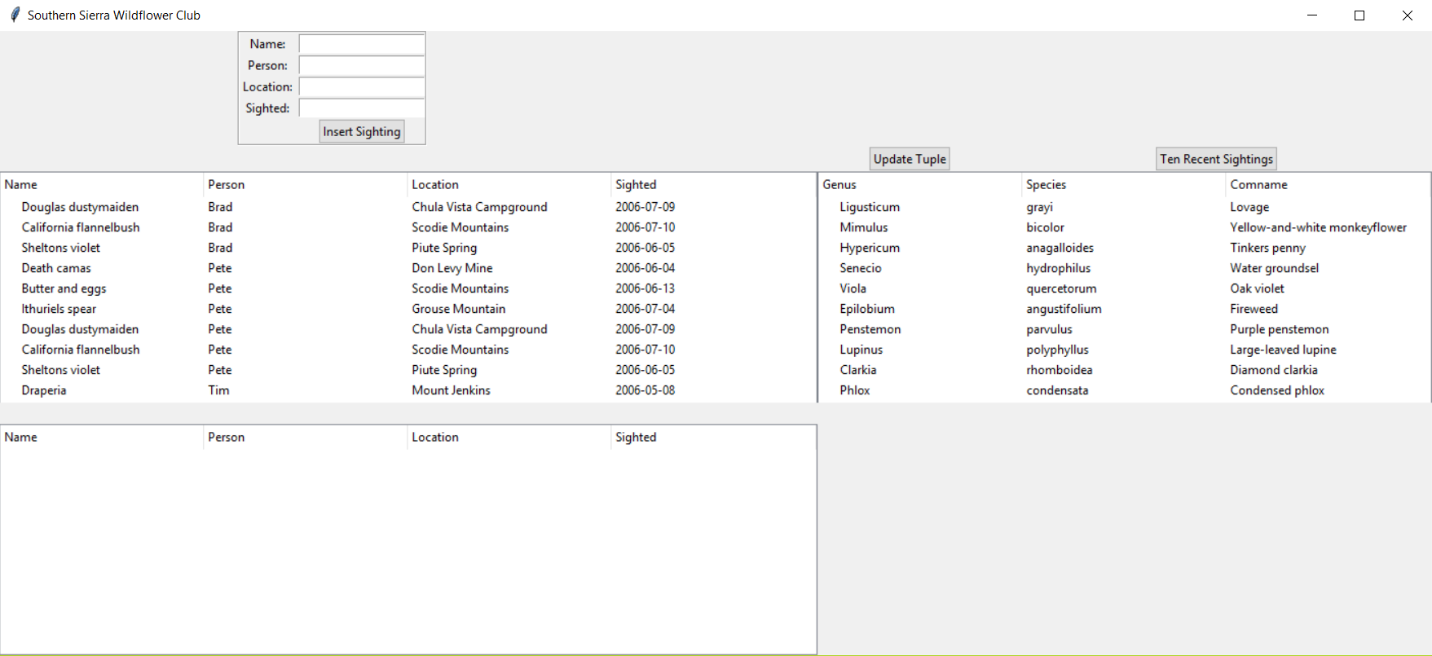
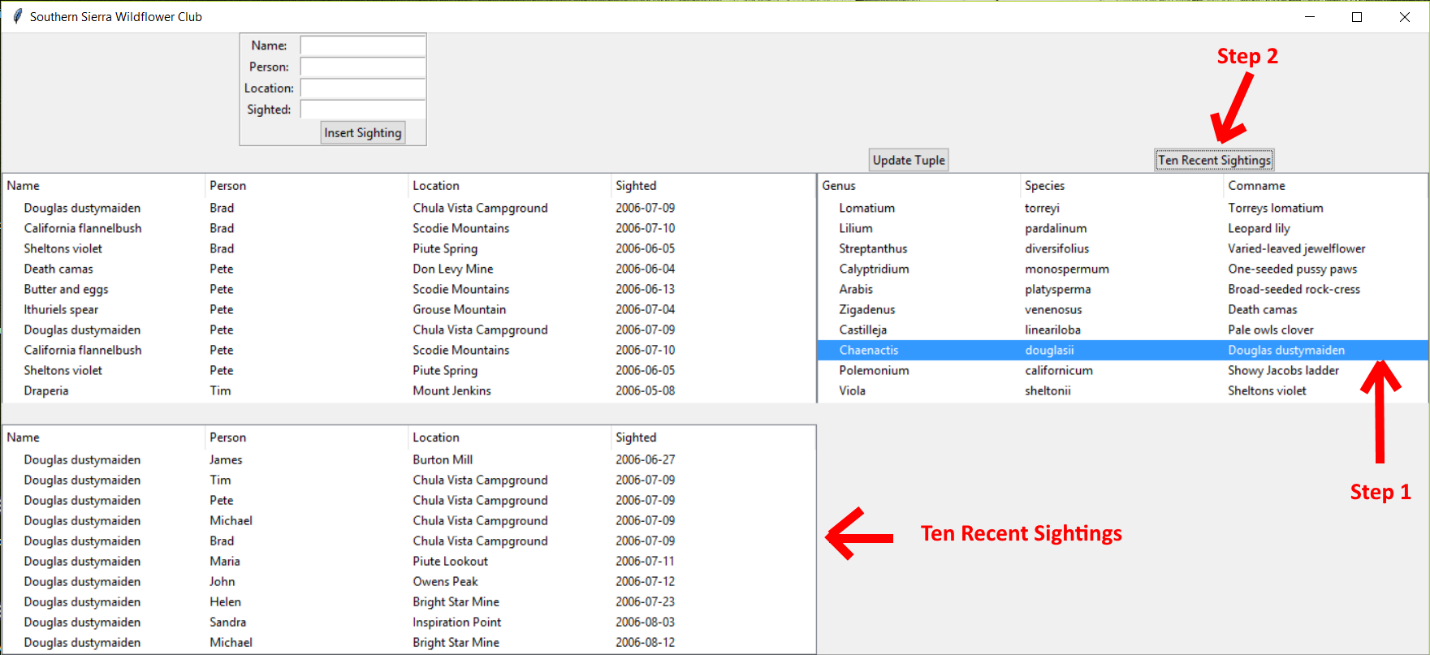
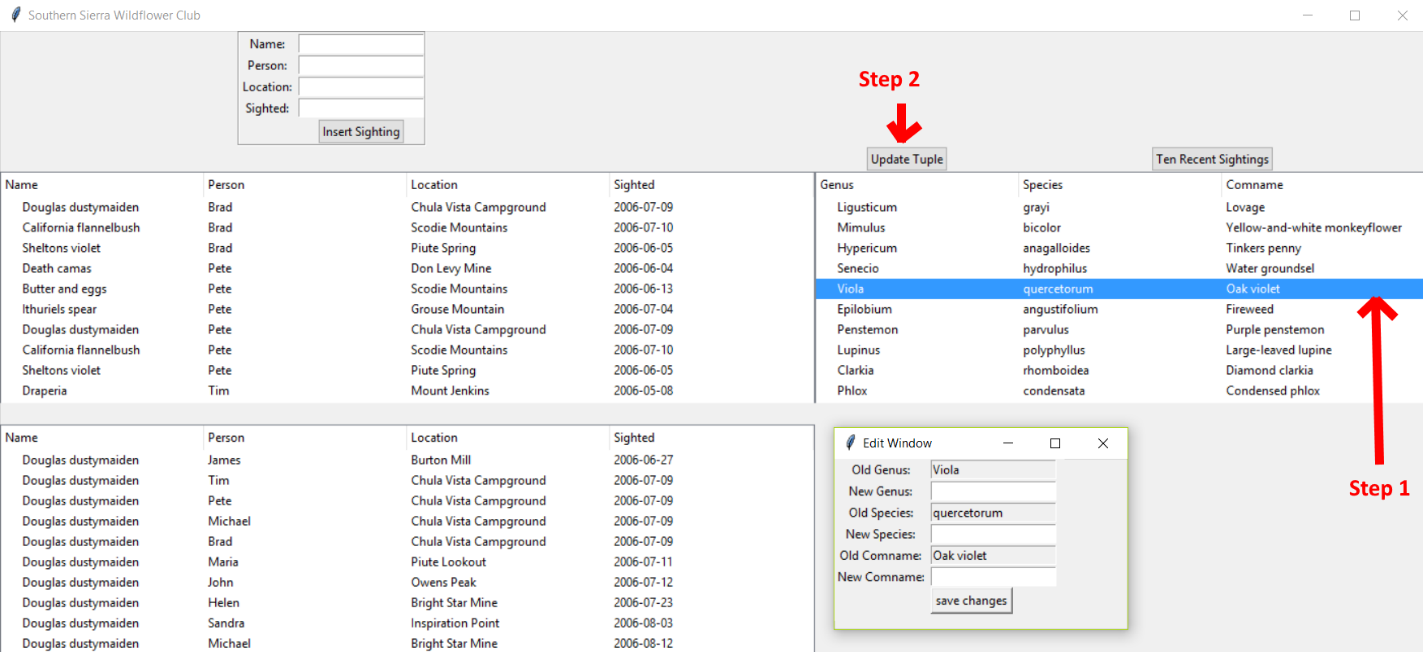
For my assignment 5 I chose to create an interface in python using the tkinter and ttk imports. This took a good deal of work first learning a bit of python, understanding all of the import features, and then figuring out all of the queries and for which table(s) to use. In order to run my interface, you first have to have python installed and then, have the “flowers.db” database file in the same folder that python is installed in. Upon opening the python file (.py) & running the interface (run -> run module) you will be greeted with the following window.



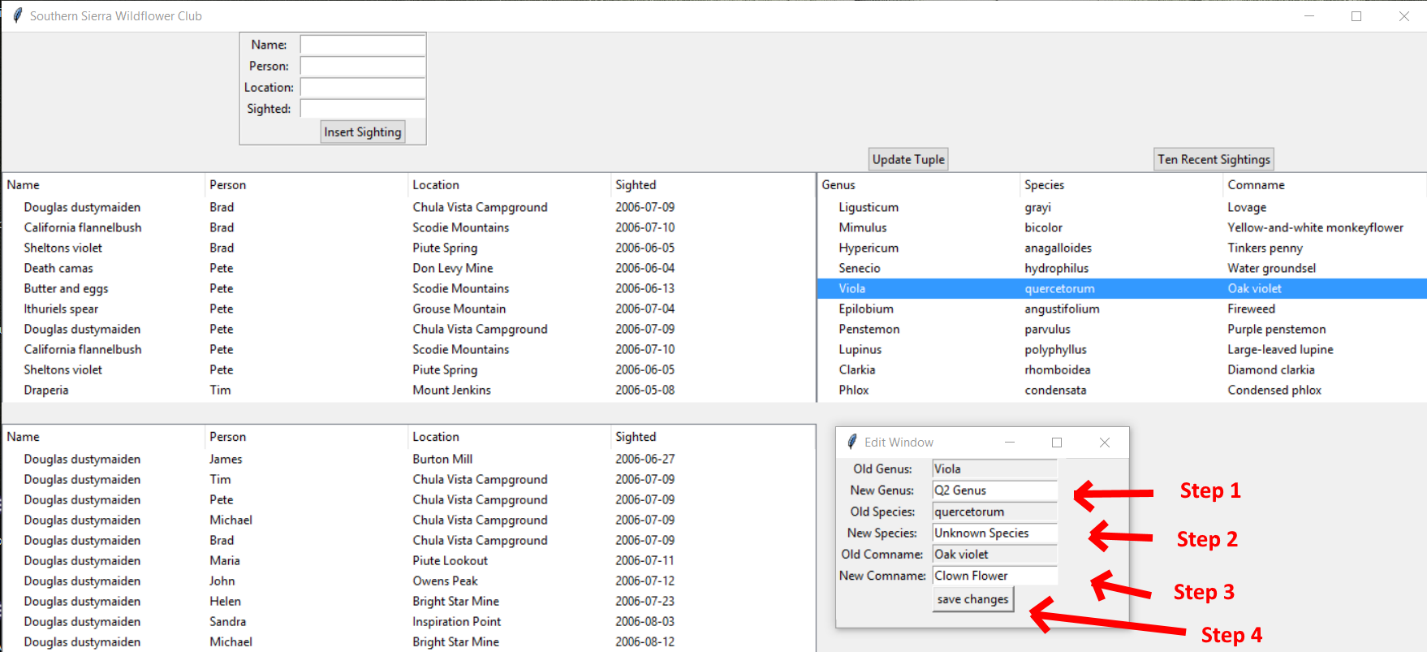
From here you have a few options that you can do to edit the database. The first option is to check the ten most recent sightings from the table of flowers. In this example we will choose Chaenactis douglasii or “Douglas dustymaiden” from the FLOWERS table and then, click Ten Recent Sightings. The resulting output is then, displayed in the below table.



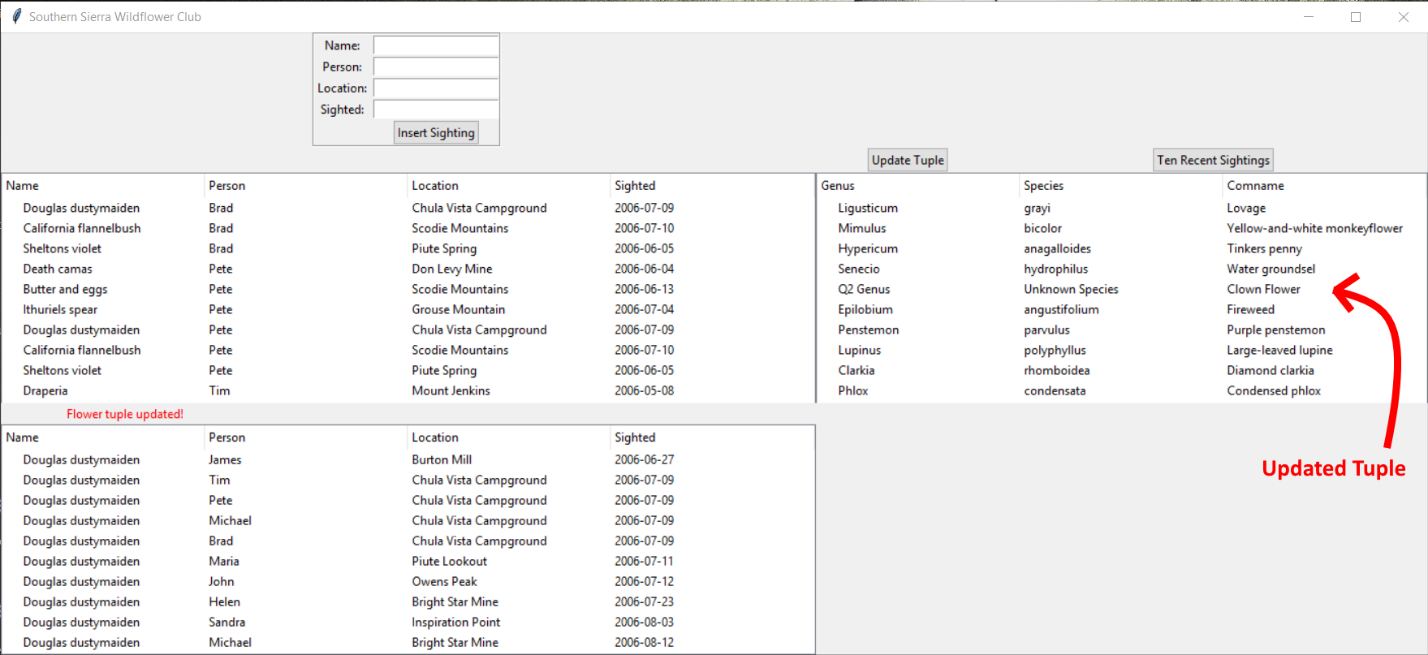
The next feature of the interface is the ability to update an already existing tuple in the FLOWERS table. In order to do this, we first select the tuple we are interested in updating. In this example we will choose “Viola quercetorum Oak violet” and then, click Update Tuple. Upon doing this, we are then greeted with a new window that allows us to enter are new input.



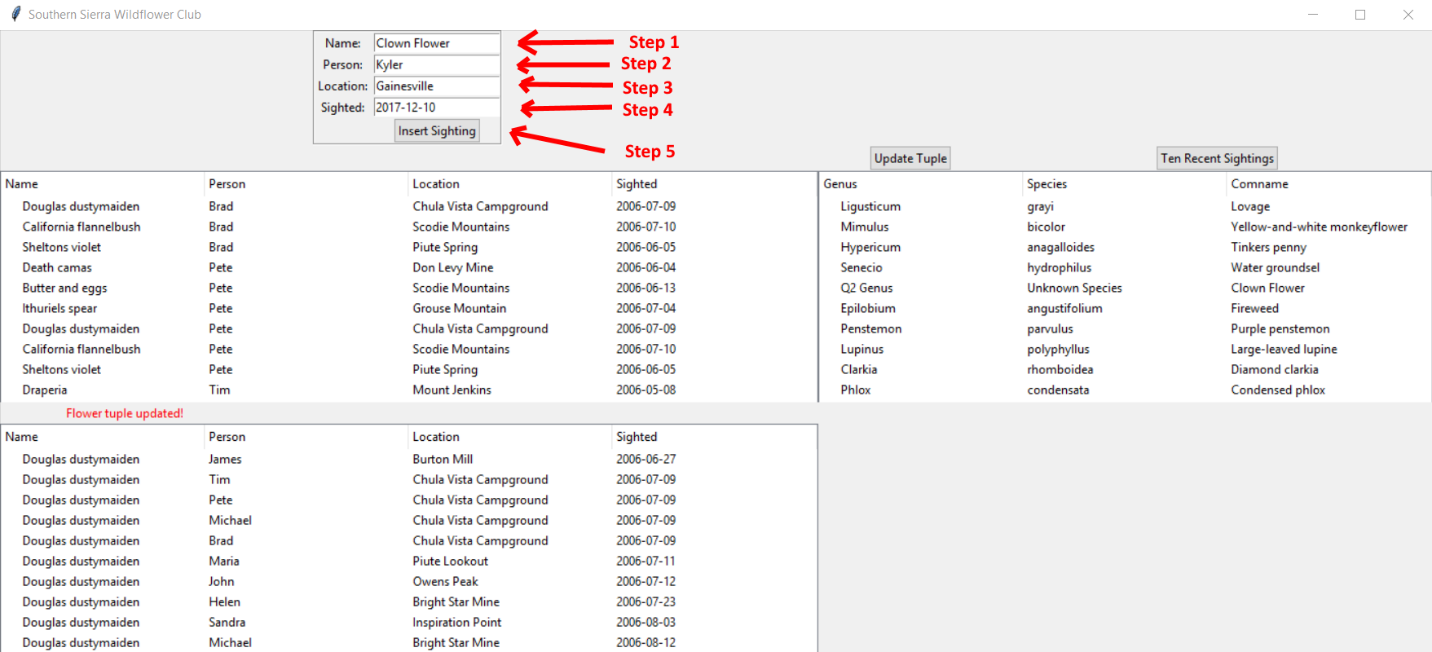
I then, enter my new genus, species, and comname information and click save changes.



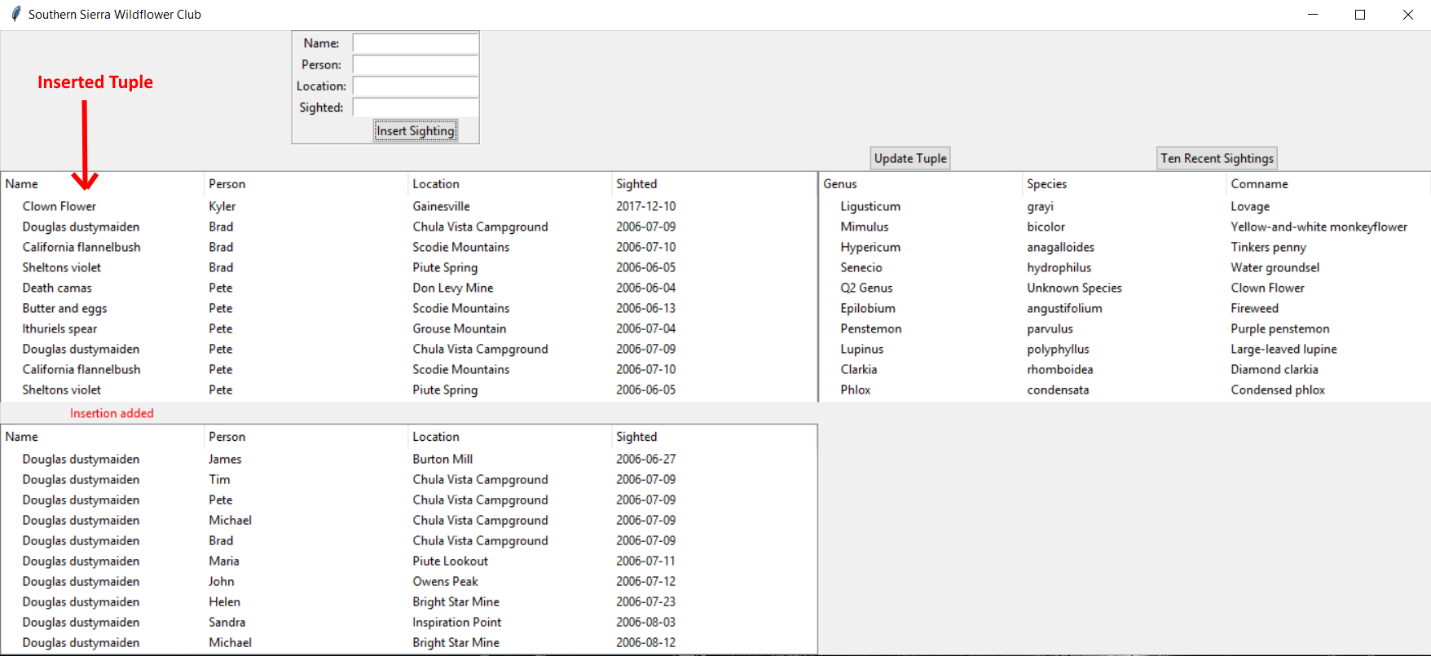
We can then, see our updated tuple with the following info.



The final ability of the interface is to insert new sightings of flowers in the SIGHTINGS table. In order to do this, simply enter the name, person, location, and sighted date in the following box & click Insert Sighting.



After entering our new input we can then, see our newly inserted tuple.



In case my python (.py) file is unable to be uploaded to canvas I will have all of the following code that created this interface below.

from tkinter import \*

from tkinter import ttk

import sqlite3

class Assignment5:

data\_base = 'flowers.db'

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

self.wind = wind;

self.wind.title('Southern Sierra Wildflower Club')

frame = LabelFrame(self.wind)

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

Label (frame, text = 'Name: ').grid(row = 1, column = 1)

self.name = Entry(frame)

self.name.grid(row = 1, column = 2)

Label (frame, text = 'Person: ').grid(row = 2, column = 1)

self.person = Entry(frame)

self.person.grid(row = 2, column = 2)

Label (frame, text = 'Location: ').grid(row = 3, column = 1)

self.location = Entry(frame)

self.location.grid(row = 3, column = 2)

Label (frame, text = 'Sighted: ').grid(row = 4, column = 1)

self.sighted = Entry(frame)

self.sighted.grid(row = 4, column = 2)

ttk.Button(frame, text = 'Insert Sighting', command = self.adding).grid(row = 5, column = 2)

self.message = Label(text = '', fg = 'red')

self.message.grid(row = 5, column = 0)

self.left\_tree = ttk.Treeview(height = 10, columns = ("person","location","sighted"))

self.left\_tree.grid(row = 2, column = 0, columnspan = 4)

self.left\_tree.heading('#0', text = 'Name', anchor = W)

self.left\_tree.heading("person", text = 'Person', anchor = W)

self.left\_tree.heading("location", text = 'Location', anchor = W)

self.left\_tree.heading("sighted", text = 'Sighted', anchor = W)

self.viewing\_records()

self.right\_tree = ttk.Treeview(height = 10, columns = ("Species","Comname"))

self.right\_tree.grid(row = 2, column = 10, columnspan = 4)

self.right\_tree.heading('#0', text = 'Genus', anchor = W)

self.right\_tree.heading("Species", text = 'Species', anchor = W)

self.right\_tree.heading("Comname", text = 'Comname', anchor = W)

self.viewing\_flower\_records()

self.lower\_tree = ttk.Treeview(height = 10, columns = ("person","location","sighted"))

self.lower\_tree.grid(row = 10, column = 0, columnspan = 4)

self.lower\_tree.heading('#0', text = 'Name', anchor = W)

self.lower\_tree.heading("person", text = 'Person', anchor = W)

self.lower\_tree.heading("location", text = 'Location', anchor = W)

self.lower\_tree.heading("sighted", text = 'Sighted', anchor = W)

ttk.Button(text = 'Update Tuple', command = self.editing).grid(row = 1, column = 10)

ttk.Button(text = 'Ten Recent Sightings', command = self.ten\_recent\_sightings).grid(row = 1, column = 12)

def run\_query (self, query, parameters = ()):

with sqlite3.connect (self.data\_base) as conn:

cursor = conn.cursor()

query\_result = cursor.execute(query, parameters)

conn.commit()

return query\_result

def viewing\_records (self):

records = self.left\_tree.get\_children()

for element in records:

self.left\_tree.delete (element)

query = 'SELECT \* FROM SIGHTINGS'

data\_base\_rows = self.run\_query(query)

for row in data\_base\_rows:

self.left\_tree.insert('' , 0, text = row[0], values = [row[1],row[2],row[3]])

def viewing\_flower\_records (self):

records = self.right\_tree.get\_children()

for element in records:

self.right\_tree.delete (element)

query = 'SELECT \* FROM FLOWERS'

data\_base\_rows = self.run\_query(query)

for row in data\_base\_rows:

self.right\_tree.insert('' , 0, text = row[0], values = [row[1],row[2]])

def adding (self):

query = 'INSERT INTO SIGHTINGS VALUES (?,?,?,?)'

parameters = (self.name.get(), self.person.get(), self.location.get(), self.sighted.get())

self.run\_query (query, parameters)

self.message ['text'] = 'Insertion added'.format (self.name.get())

self.name.delete (0, END)

self.person.delete (0, END)

self.location.delete (0, END)

self.sighted.delete (0, END)

self.viewing\_records()

def update\_records (self, new\_genus, old\_genus, new\_species, old\_species, new\_comname, old\_comname):

query = 'UPDATE FLOWERS SET genus = ?, species = ?, comname = ? WHERE genus = ? AND species = ? AND comname = ?'

parameters = (new\_genus, new\_species, new\_comname, old\_genus, old\_species, old\_comname)

self.run\_query (query, parameters)

self.edit\_wind.destroy()

self.message['text'] = 'Flower tuple updated!'

self.viewing\_flower\_records()

def editing (self):

self.message['text'] = ''

try:

self.right\_tree.item (self.right\_tree.selection())['values'][0]

except IndexError as e:

self.message['text'] = 'Select a flower'

return

old\_genus = self.right\_tree.item(self.right\_tree.selection())['text']

old\_species = self.right\_tree.item (self.right\_tree.selection())['values'][0]

old\_comname = self.right\_tree.item (self.right\_tree.selection())['values'][1]

self.edit\_wind = Toplevel()

self.edit\_wind.title('Edit Window')

Label (self.edit\_wind, text = 'Old Genus: ').grid(row = 0, column = 1)

Entry (self.edit\_wind, textvariable = StringVar(self.edit\_wind, value = old\_genus), state = 'readonly').grid(row = 0, column = 2)

Label (self.edit\_wind, text = 'New Genus: ').grid(row = 1, column = 1)

new\_genus = Entry(self.edit\_wind)

new\_genus.grid(row = 1, column = 2)

Label (self.edit\_wind, text = 'Old Species: ').grid(row = 2, column = 1)

Entry (self.edit\_wind, textvariable = StringVar(self.edit\_wind, value = old\_species), state = 'readonly').grid(row = 2, column = 2)

Label (self.edit\_wind, text = 'New Species: ').grid(row = 3, column = 1)

new\_species = Entry(self.edit\_wind)

new\_species.grid(row = 3, column = 2)

Label (self.edit\_wind, text = 'Old Comname: ').grid(row = 4, column = 1)

Entry (self.edit\_wind, textvariable = StringVar(self.edit\_wind, value = old\_comname), state = 'readonly').grid(row = 4, column = 2)

Label (self.edit\_wind, text = 'New Comname: ').grid(row = 5, column = 1)

new\_comname = Entry(self.edit\_wind)

new\_comname.grid(row = 5, column = 2)

Button(self.edit\_wind, text = 'save changes', command = lambda:self.update\_records(new\_genus.get(), old\_genus, new\_species.get(), old\_species, new\_comname.get(), old\_comname)).grid(row = 6, column = 2, sticky = W)

self.edit\_wind.mainloop()

def ten\_recent\_sightings (self):

try:

self.right\_tree.item (self.right\_tree.selection())['values'][0]

except IndexError as e:

self.message['text'] = 'Select a flower'

return

selected\_flower = self.right\_tree.item (self.right\_tree.selection())['values'][1]

query = 'SELECT \* FROM SIGHTINGS WHERE name = (SELECT comname FROM FLOWERS WHERE comname = ?) ORDER BY sighted DESC LIMIT 10'

parameters = (selected\_flower)

data\_base\_rows = self.run\_query (query, (parameters,))

records = self.lower\_tree.get\_children()

for element in records:

self.lower\_tree.delete (element)

for row in data\_base\_rows:

self.lower\_tree.insert('' , 0, text = row[0], values = [row[1],row[2],row[3]])

if \_\_name\_\_ == '\_\_main\_\_':

wind = Tk ()

appliation = Assignment5 (wind)

wind.mainloop()