Johnny Li
CIS4301
12/12/2019
Group Member: David Li

Assignment 5

Extra Credit- Aesthetically pleasing

A GUI was implemented rather than a text-based input/output interface for improved user interaction and viewing. This GUI was made for only the original functionality of the assignment: query, insert, and update. The GUI contain interactive buttons and directions to guide the user.

Image EC.1: SSWC Main Menu

Query- Allow the user to select from a list of flowers. Using the selected flower, display the 10 most recent sightings of the selected flower. Information should include the date, location, and who sighted the flower.

Upon clicking the "Query" button, the user will be prompted with the screen as shown below:

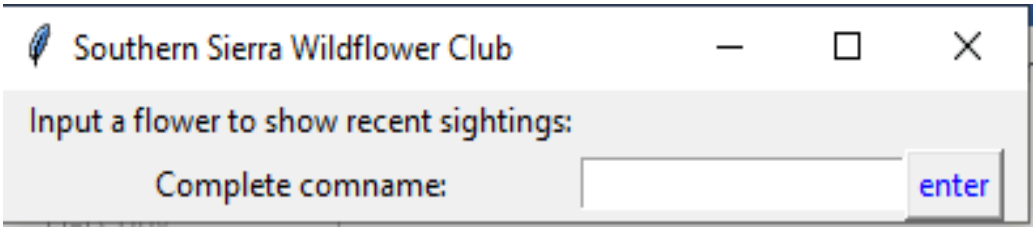

Image Q.1: Query Input Screen

When the user input the common name of a flower from the known list of 'comname' in the "SIGHTING" table and press enter they will be shown the 10 most recent sightings of the selected flower. Information includes the date (from most recent to oldest), location, and who sighted the flower as shown below:

Image Q.2: Example California flannelbush

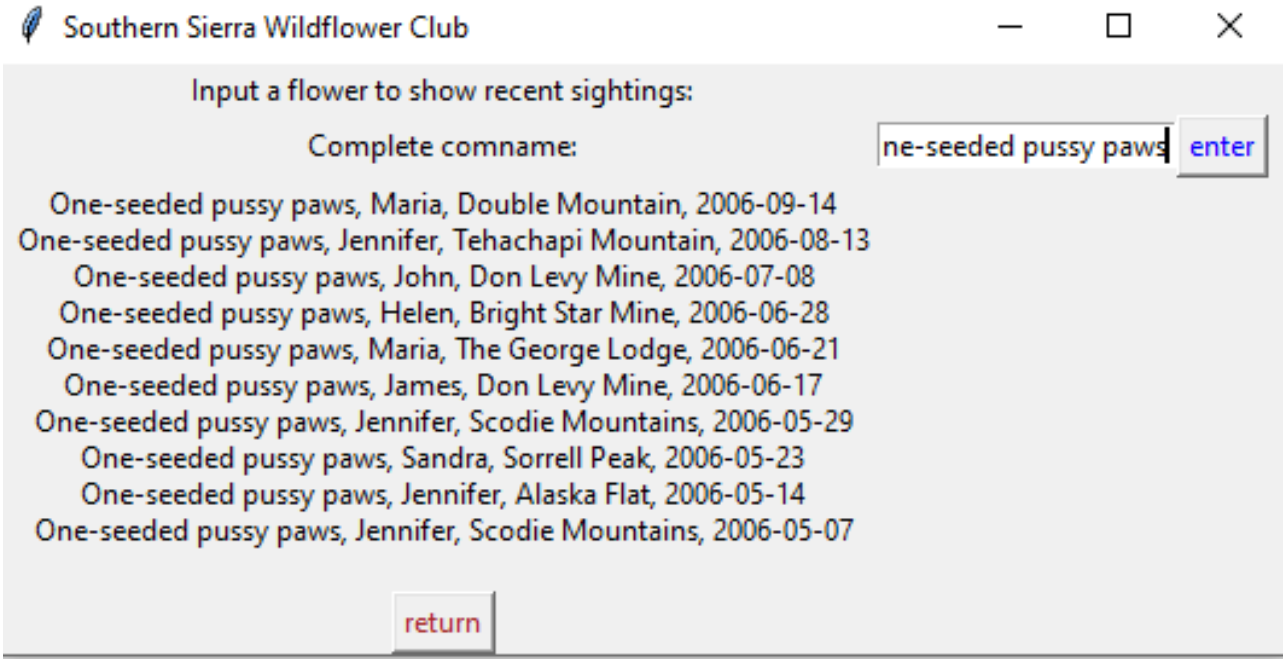

Image Q.3: Example One-seeded pussy paws

The user can type in a new common name in the input box, on the top right, and press enter to generate a new list of recent sightings of the selected flower or click the return button down below to go back to the Main menu.

Insert- Allow a user to insert a new sighting of a flower.

Upon clicking the “Insert” button, the user will be prompted with the screen as shown below:

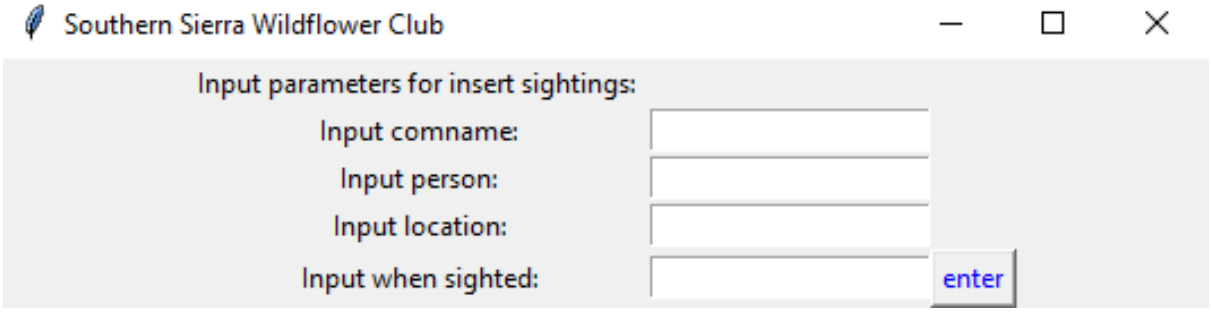

Image I.1: Insert Input Screen

User can input the following information: common name of a flower, person who sighted the flower, the location of the flower, and the date (YYYY-MM-DD) it was sighted or leave any part blank to result in a null input case. Upon pressing enter it will insert the new sighting to the “SIGHTING” table. This is shown below:

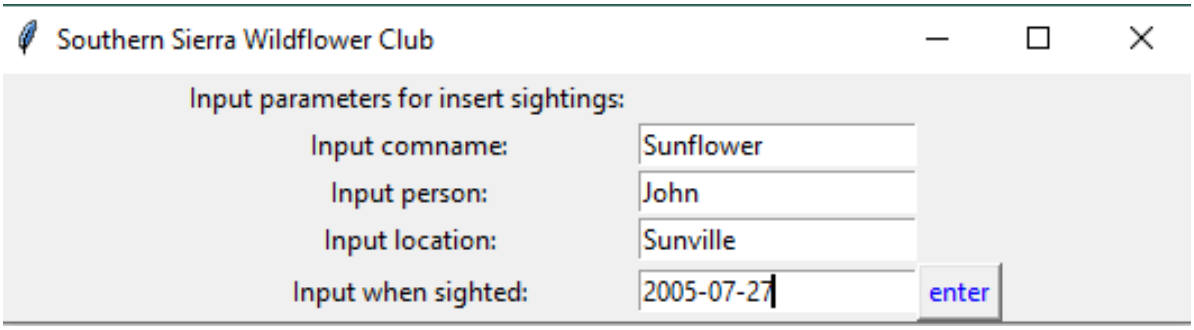

Image I.2: Example Sunflower

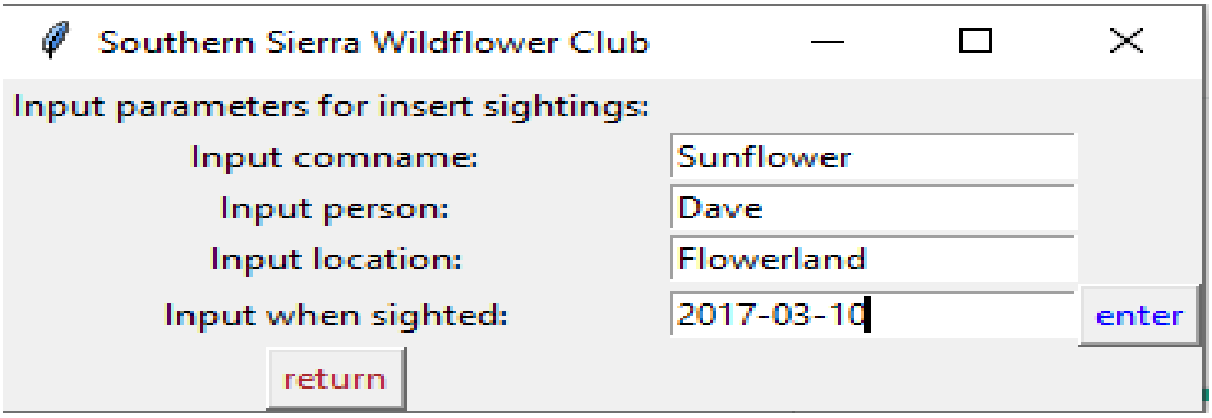

Image I.3: Example Sunflower with different information

Image I.4: Example Birds of Paradise

The user can type in new information in the input boxes and press enter to insert a new sighting or click the return button down below to go back to the Main menu.

In order to view the newly inserted sightings, the user would have go to the "Query" function and type in the submitted common name shown below:

Image I.5: Example Sunflower Result

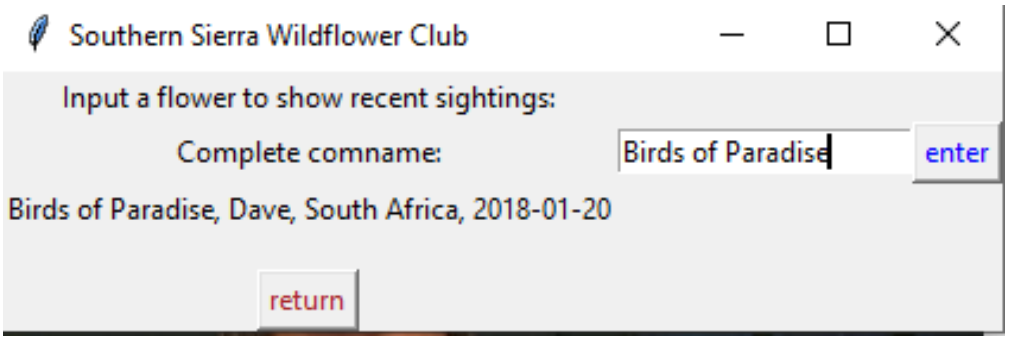

Image I.6: Example Birds of Paradise Result

Update- Allow a user to select and update flower information.

Upon clicking the "Update" button, the user will be prompted with the screen as shown below:

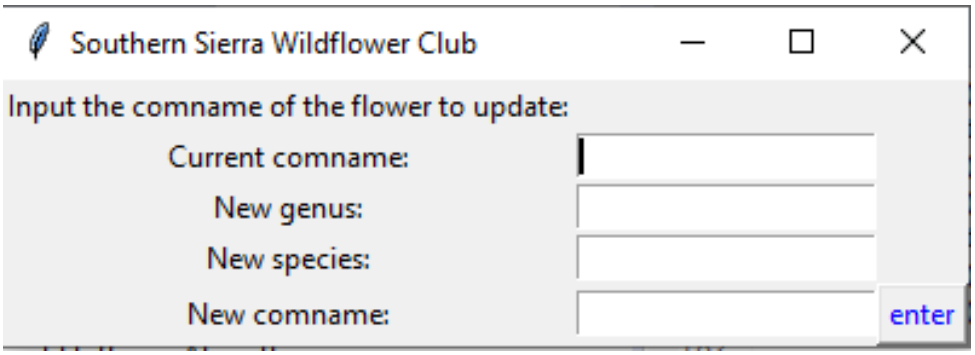

Image U.1: Update Input Screen

The user must input the ‘Current comname’ and can input following information: new common name of the flower, new genus of the flower, and new species of the flower to be updated or leave any part blank to result in a null input case. Upon pressing enter it will update the information in the “FLOWER” table. This is shown below:

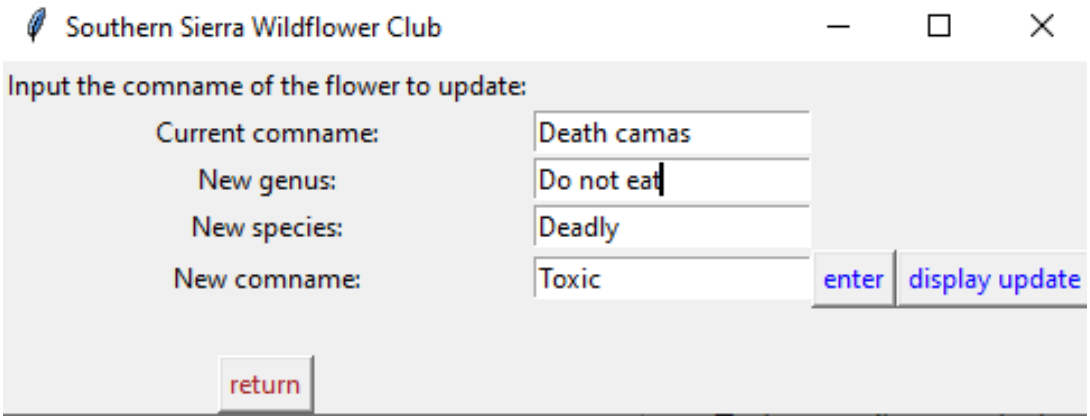

Image U.2: Example Death camas

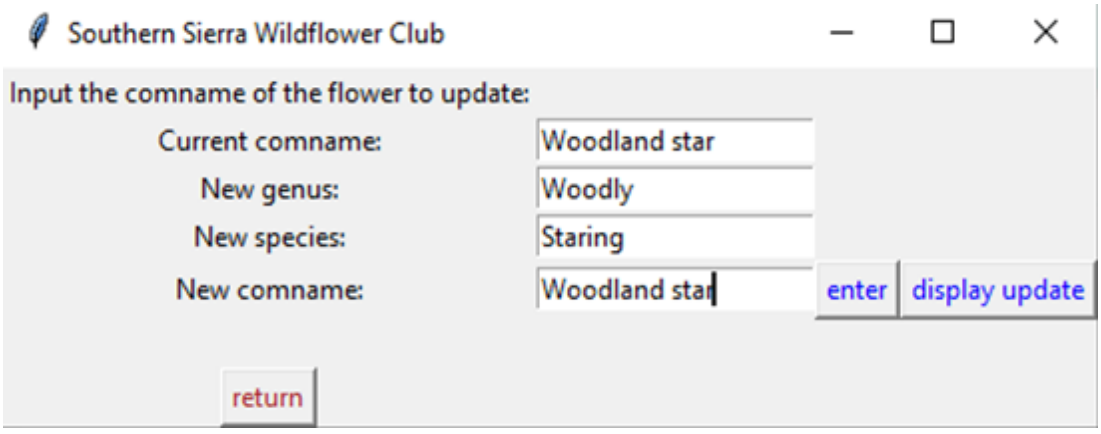

Image U.3: Example Woodland star

The user can type in new information in the input boxes and press enter to update a flower, click the return button down below to go back to the Main menu, or press the display update in the far right to see the changes made to the “FLOWER” table (genus, species, comname) as shown below:

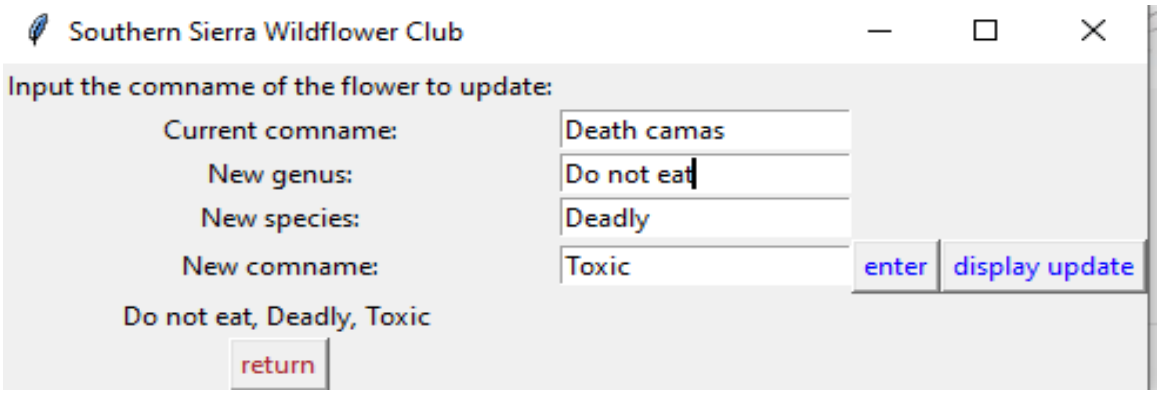

Image U.4: Example Death camas Result

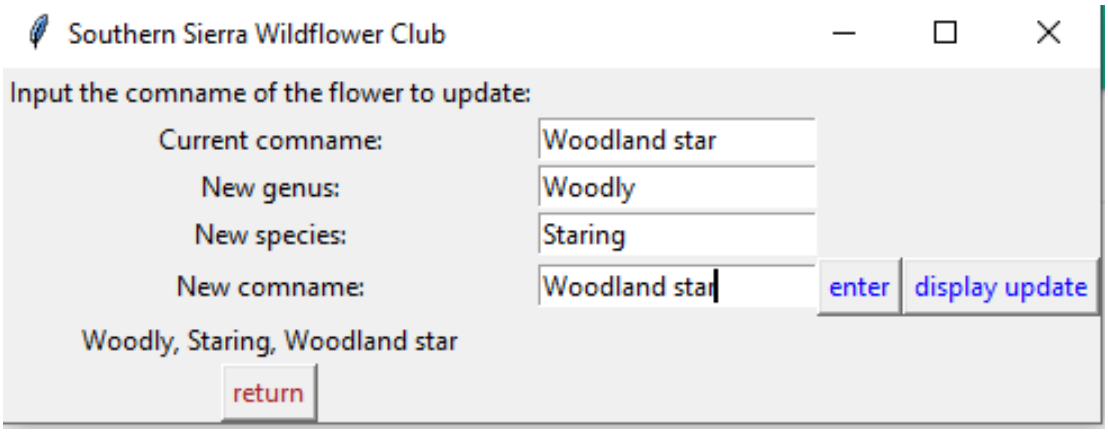

Image U.5: Example Woodland star Result

Note: To be able to access the GUI, the user must have the tkinter module installed in python.

Code- For python.

```
#Author: Johnny Li and David Li
#Assignment 5 (Flowers)

#References
#https://nitratine.net/blog/post/python-sqlite3-basics/
#https://www.youtube.com/watch?v=SQj17D1Q_6s Python SQLite Basics
#https://bytes.com/topic/python/answers/44281-getting-sender-widgets-name-function-
tkinter
import sqlite3
connection = sqlite3.connect('flowers2019.db') #Load database
cursor = connection.cursor()

from tkinter import*    #GUI library

#Title of UI
root = Tk()
root.title('Southern Sierra Wildflower Club')

#Setup UI
topFrame = Frame(root)
topFrame.pack()
topFrame2 = Frame(root)
topFrame3 = Frame(root)
topFrame4 = Frame(root)
topFrame5 = Frame(root)

s=""    #Input string

#Main menu option
def mainoption(event):
    global s    #Make string global
    s=event.widget['text']  #Load text of button
    topFrame.pack_forget()  #Reset Frame
    print(s)    #Testing: Print selection
    #Selection:
    if s == "Query":
        topFrame2.pack()
    if s == "Update":
        topFrame3.pack()
    if s == "Insert":
        topFrame4.pack()
    if s == "Exit":
        root.destroy()  #Destroy everything

#Selection: Query
def mainoption2(event):
    #SQL code
    cursor.execute(
        '''SELECT *
        FROM SIGHTINGS
        where name = "'''+ e1.get() + '\"'+'''
```

```
        ORDER BY sighted DESC
        LIMIT 10''')
    row = cursor.fetchone() #Load table result
    sightingStr = ""    #Sighting string
    for values in row:  #For loop
        sightingStr+=values+", "    #Format Output
    sightingStr=sightingStr[:-2]
    sightingStr+="\n"
    row = cursor.fetchone()
    while row is not None:  #While loop full list
        for values in row:  #For loop
            sightingStr+=values+", "    #Format Output
        sightingStr=sightingStr[:-2]
        sightingStr+="\n"
        row = cursor.fetchone()
    connection.commit()
    Label(topFrame2, text=sightingStr).grid(row=2)
    #Return button
    button5 = Button(topFrame2, text="return", fg = "brown")
    button5.grid(row=3)
    button5.bind("<Button-1>", mainoption3)

#Reset everything
def mainoption3(event):
    topFrame4.pack_forget()
    topFrame3.pack_forget()
    topFrame2.pack_forget()
    topFrame.pack()

#Display update button
def displayupdate(newName):
    strResult=""    #Update string
    #SQL code
    cursor.execute(
          '''SELECT *
        FROM FLOWERS
        where comname = "'''+ newName + '\"')
    row = cursor.fetchone()
    while row is not None:   #While loop full list
        for values in row:  #For loop
            strResult+=values+", "  #Format Output
        strResult=strResult[:-2]
        row = cursor.fetchone()
    connection.commit()
    Label(topFrame3, text=strResult).grid(row=5)

#Selection: Update
def mainoption4(event):
    count = 0   #Comma count
    s=""
    #New information string variable
    newGenus=""
    newSpecies=""
    newComname = ""
    if e3.get():
```

```
        s+="genus"
    if e4.get():
        s+="species"
    if e5.get():
        s+="comname"
    #Check if value exist
    if "genus" in s:
        count = count+1
    if "species" in s:
        count = count+1
    if "comname" in s:
        count = count+1
    #Format SQL code
    if "genus" in s:
        newGenus = "genus = \'"+ e3.get() +"\'"
        count = count -1
        if count != 0:
            newGenus += ", "
    if "species" in s:
        newSpecies = "species = \'"+ e4.get() +"\'"
        count = count -1
        if count != 0:
            newSpecies += ", "
    if "comname" in s:
        newComname = "comname = \'"+ e5.get() +"\'"
    strCommand = '''UPDATE FLOWERS
    SET '''+newGenus+newSpecies+newComname+ " WHERE comname = "+ '"' +e2.get() + '"';
    print(strCommand)   #Testing: Print SQL code
    cursor.execute(strCommand)
    row = cursor.fetchone()
    while row is not None:  #While loop
        row = cursor.fetchone()
    connection.commit()
    curName ="" #Current name string
    if(not e5.get()):
        curName=e2.get()    #New common name
    else:
        curName=e5.get()    #Old common name
    print(curName) #Testing: Print current common name
    #Display Update button
    button6 = Button(topFrame3, text="display update", fg = "blue")
    button6.grid(row=4, column = 3)
    button6.bind("<Button-1>", lambda event, a=curName: displayupdate(a))
    #Return button
    button5 = Button(topFrame3, text="return", fg = "brown")
    button5.grid(row=6)
    button5.bind("<Button-1>", mainoption3)

#Selection: Insert
def mainoption5(event):
    #Format SQL
    newComname = "\""+e6.get()+"\", "
    newPerson = "\""+e7.get()+"\", "
    newLocation = "\""+e8.get()+"\", "
    newSighted = "\""+e9.get()+"\" "
```

```
    #SQL code
    cursor.execute(
    "INSERT INTO SIGHTINGS VALUES
"+"("+newComname+newPerson+newLocation+newSighted+")")
    connection.commit()
    #Return button
    button5 = Button(topFrame4, text="return", fg = "brown")
    button5.grid(row=5)
    button5.bind("<Button-1>", mainoption3)

#Main menu text coloring
theLabel = Label(topFrame, text ="Select one:")
theLabel.pack(side = TOP)
button1 = Button(topFrame, text="Query", fg="red")
button1.bind("<Button-1>", mainoption)
button1.pack(side = LEFT )
button2 = Button(topFrame, text="Update", fg="blue")
button2.bind("<Button-1>", mainoption)
button2.pack(side = LEFT )
button3 = Button(topFrame, text="Insert", fg="green")
button3.bind("<Button-1>", mainoption)
button3.pack(side = LEFT )
button7 = Button(topFrame, text="Exit", fg = "brown")
button7.bind("<Button-1>", mainoption)
button7.pack(side = LEFT)

#Query menu text coloring
theLabel = Label(topFrame2, text ="Input a flower to show recent
sightings:").grid(row=0)
Label(topFrame2, text='Current comname:').grid(row=1)
comname = StringVar()
e1 = Entry(topFrame2, textvariable = comname)
e1.grid(row=1, column=1)

#https://stackoverflow.com/questions/1101750/tkinter-attributeerror-nonetype-object-
has-no-attribute-attribute-name
#can't put .grid for button cause a.() b.() and grid returning none
button5 = Button(topFrame2, text="enter", fg = "blue")
button5.grid(row=1, column = 2)

#https://stackoverflow.com/questions/7299955/tkinter-binding-a-function-with-
arguments-to-a-widget
#button5.bind("<Button-1>", lambda event, a=comname.get(): mainoption2(a))
#https://www.youtube.com/watch?v=qCnBkZLb-E4
button5.bind("<Button-1>", mainoption2)

#Update menu text coloring
theLabel = Label(topFrame3, text ="Input the comname of the flower to
update:").grid(row=0)
Label(topFrame3, text='Current comname:').grid(row=1)
comname = StringVar()
e2 = Entry(topFrame3, textvariable = comname)
e2.grid(row=1, column=1)
Label(topFrame3, text='New genus:').grid(row=2)
e3 = Entry(topFrame3)
```

```
e3.grid(row=2, column=1)
Label(topFrame3, text='New species:').grid(row=3)
e4 = Entry(topFrame3)
e4.grid(row=3, column=1)
Label(topFrame3, text='New comname:').grid(row=4)
e5 = Entry(topFrame3)
e5.grid(row=4, column=1)
button5 = Button(topFrame3, text="enter", fg = "blue")
button5.grid(row=4, column = 2)
button5.bind("<Button-1>", mainoption4)

#Sighting menu text coloring
theLabel = Label(topFrame4, text ="Input parameters for insert sightings:
").grid(row=0)
Label(topFrame4, text='Input comname:').grid(row=1)
e6 = Entry(topFrame4)
e6.grid(row=1, column=1)
Label(topFrame4, text='Input person:').grid(row=2)
e7 = Entry(topFrame4)
e7.grid(row=2, column=1)
Label(topFrame4, text='Input location:').grid(row=3)
e8 = Entry(topFrame4)
e8.grid(row=3, column=1)
Label(topFrame4, text='Input when sighted:').grid(row=4)
e9 = Entry(topFrame4)
e9.grid(row=4, column=1)
button6 = Button(topFrame4, text="enter", fg = "blue")
button6.grid(row=4, column = 2)
button6.bind("<Button-1>", mainoption5)

root.mainloop() #Infinite loop
```