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.

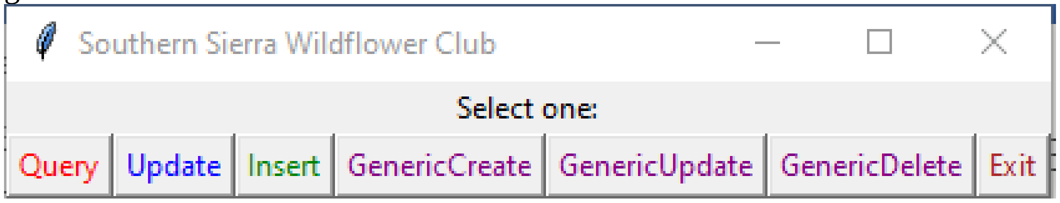


Image EC.1: SSWC Main Menu

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:

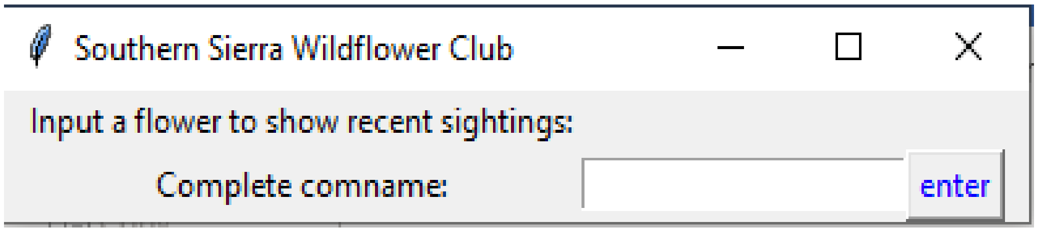
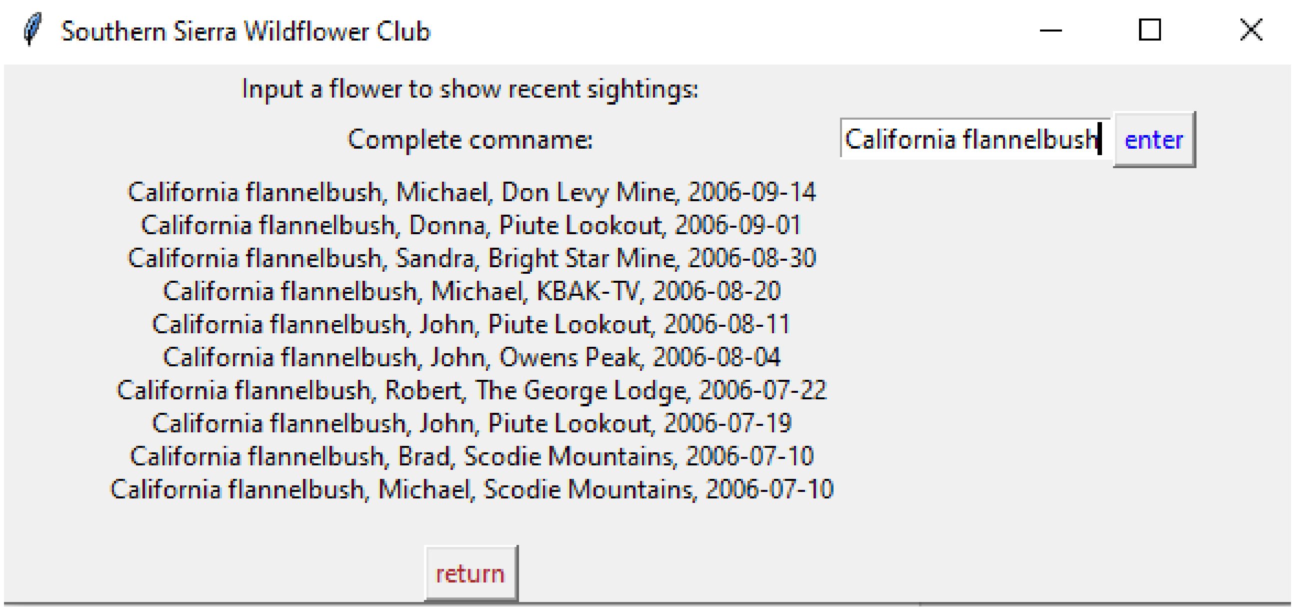

Southern Sierra Wildflower Club
Input a flower to show recent sightings:
Complete comname:
enter

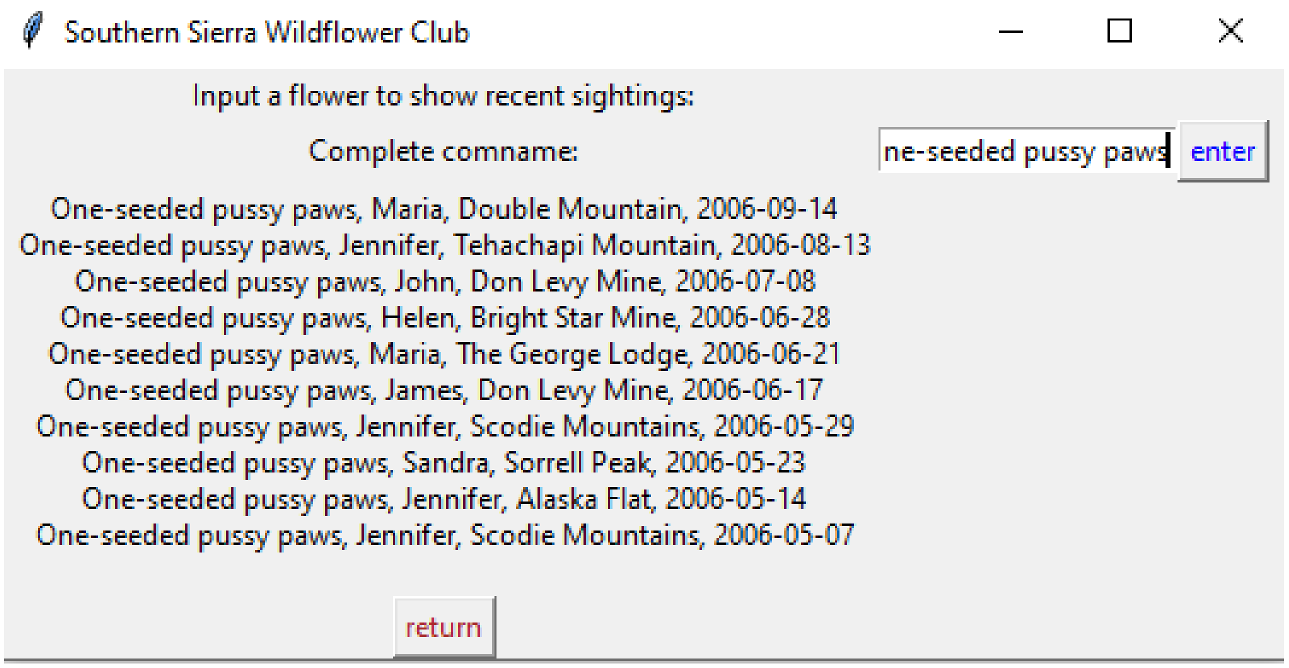
Image Q.1: Query Input Screen

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:


Southern Sierra Wildflower Club
Input a flower to show recent sightings:
Complete comname:
California flannelbush
enter
return

```
California flannelbush, Michael, Don Levy Mine, 2006-09-14
California flannelbush, Donna, Piute Lookout, 2006-09-01
California flannelbush, Sandra, Bright Star Mine, 2006-08-30
California flannelbush, Michael, KBAK-TV, 2006-08-20
California flannelbush, John, Piute Lookout, 2006-08-11
California flannelbush, John, Owens Peak, 2006-08-04
California flannelbush, Robert, The George Lodge, 2006-07-22
California flannelbush, John, Piute Lookout, 2006-07-19
California flannelbush, Brad, Scodie Mountains, 2006-07-10
California flannelbush, Michael, Scodie Mountains, 2006-07-10
```

Image Q.2: Example California flannelbush


Southern Sierra Wildflower Club
Input a flower to show recent sightings:
Complete comname:
ne-seeded pussy paws
enter
return

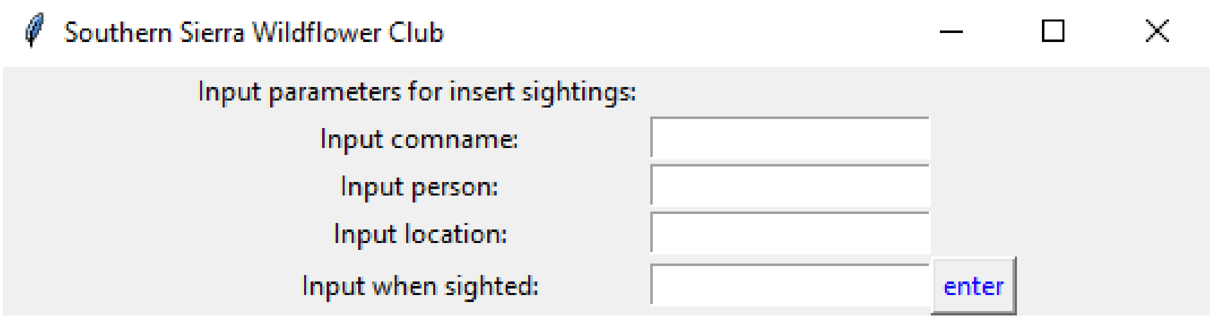
```
One-seeded pussy paws, Maria, Double Mountain, 2006-09-14
One-seeded pussy paws, Jennifer, Tehachapi Mountain, 2006-08-13
One-seeded pussy paws, John, Don Levy Mine, 2006-07-08
One-seeded pussy paws, Helen, Bright Star Mine, 2006-06-28
One-seeded pussy paws, Maria, The George Lodge, 2006-06-21
One-seeded pussy paws, James, Don Levy Mine, 2006-06-17
One-seeded pussy paws, Jennifer, Scodie Mountains, 2006-05-29
One-seeded pussy paws, Sandra, Sorrell Peak, 2006-05-23
One-seeded pussy paws, Jennifer, Alaska Flat, 2006-05-14
One-seeded pussy paws, Jennifer, Scodie Mountains, 2006-05-07
```

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:



Southern Sierra Wildflower Club

Input parameters for insert sightings:

Input comname:

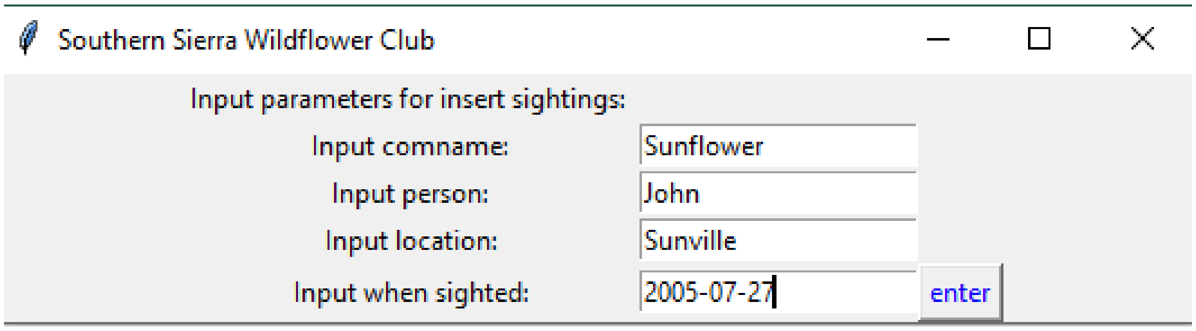
Input person:

Input location:

Input when sighted:

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:



Southern Sierra Wildflower Club

Input parameters for insert sightings:

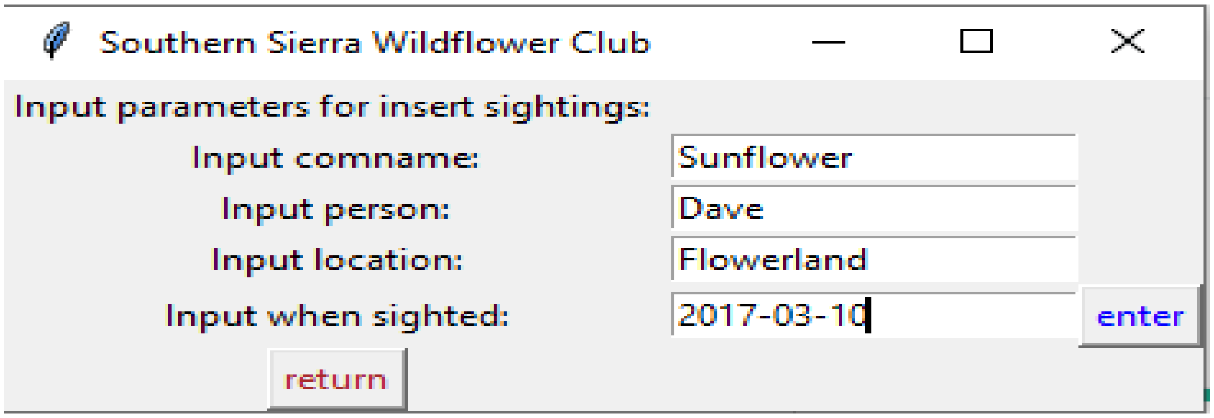
Input comname: Sunflower

Input person: John

Input location: Sunville

Input when sighted: 2005-07-27

Image I.2: Example Sunflower



Southern Sierra Wildflower Club

Input parameters for insert sightings:

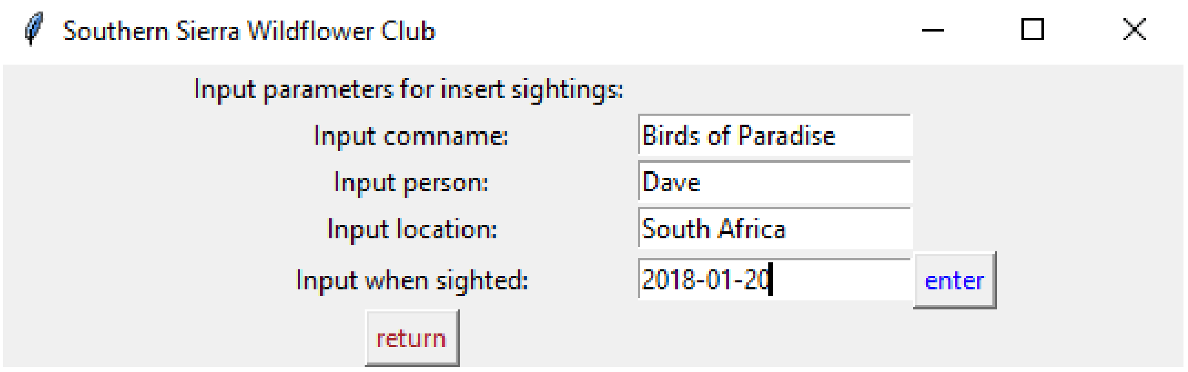
Input comname: Sunflower

Input person: Dave

Input location: Flowerland

Input when sighted: 2017-03-10

Image I.3: Example Sunflower with different information


Southern Sierra Wildflower Club

Input parameters for insert sightings:

Input comname: Birds of Paradise

Input person: Dave

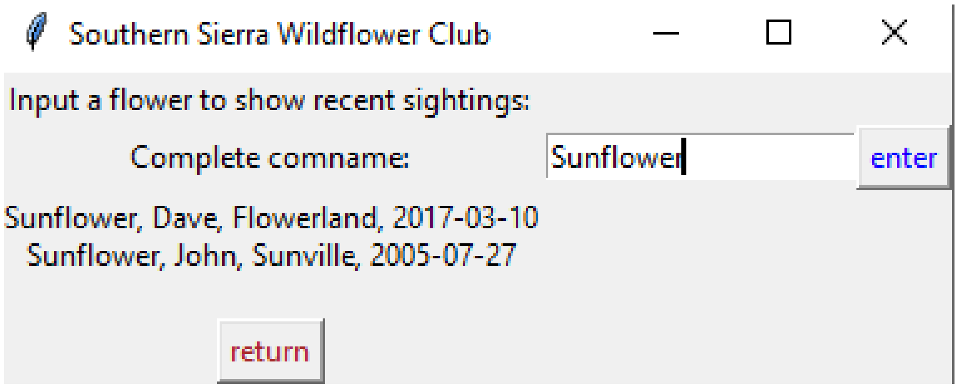
Input location: South Africa

Input when sighted: 2018-01-20

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:

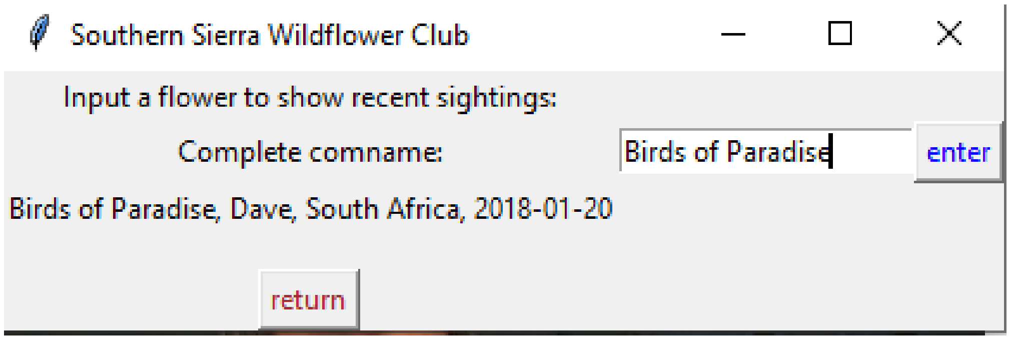

Southern Sierra Wildflower Club

Input a flower to show recent sightings:

Complete comname: Sunflower

Sunflower, Dave, Flowerland, 2017-03-10
Sunflower, John, Sunville, 2005-07-27

Image I.5: Example Sunflower Result


Southern Sierra Wildflower Club

Input a flower to show recent sightings:

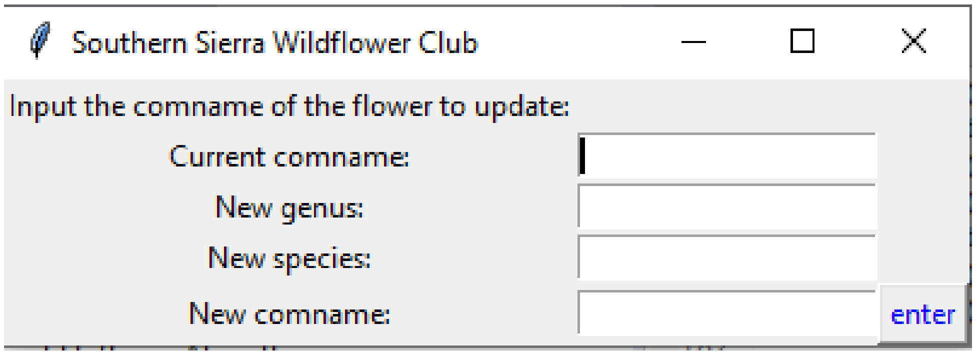
Complete comname: Birds of Paradise

Birds of Paradise, Dave, South Africa, 2018-01-20

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:

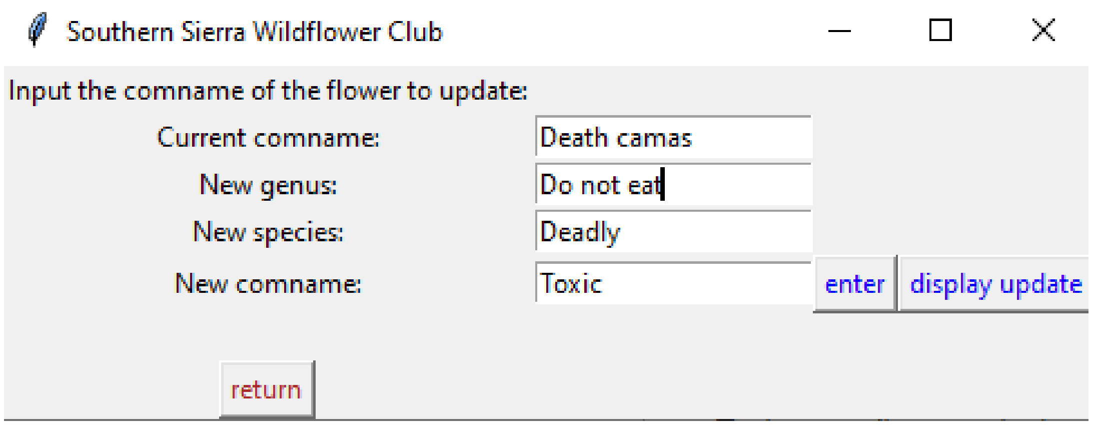

Southern Sierra Wildflower Club

Input the comname of the flower to update:

Current comname:	
New genus:	
New species:	
New comname:	

Image U.1: Update Input Screen

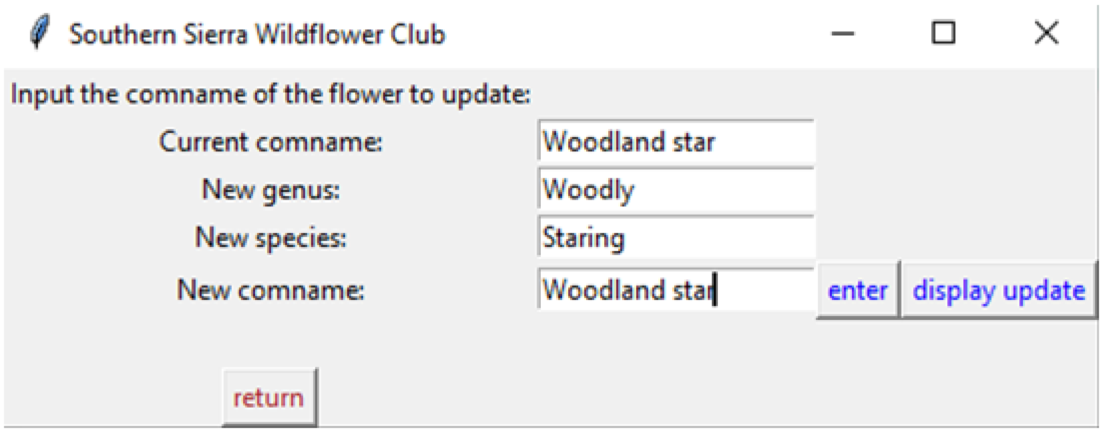
The user must input the ‘Current comname’ and can input following information: new common name of the flower, new genus of the flower, and new species of the flower to be updated or leave any part blank to result in a null input case. Upon pressing enter it will update the information in the “FLOWER”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


Southern Sierra Wildflower Club

Input the comname of the flower to update:

Current comname:	Death camas
New genus:	Do not eat
New species:	Deadly
New comname:	Toxic

Image U.2: Example Death camas

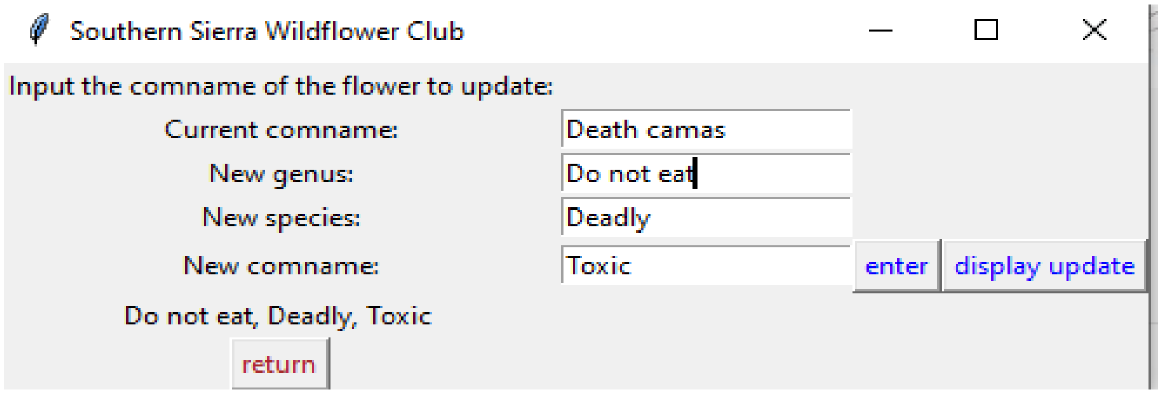

Southern Sierra Wildflower Club

Input the comname of the flower to update:

Current comname:	Woodland star
New genus:	Woodly
New species:	Staring
New comname:	Woodland star

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:

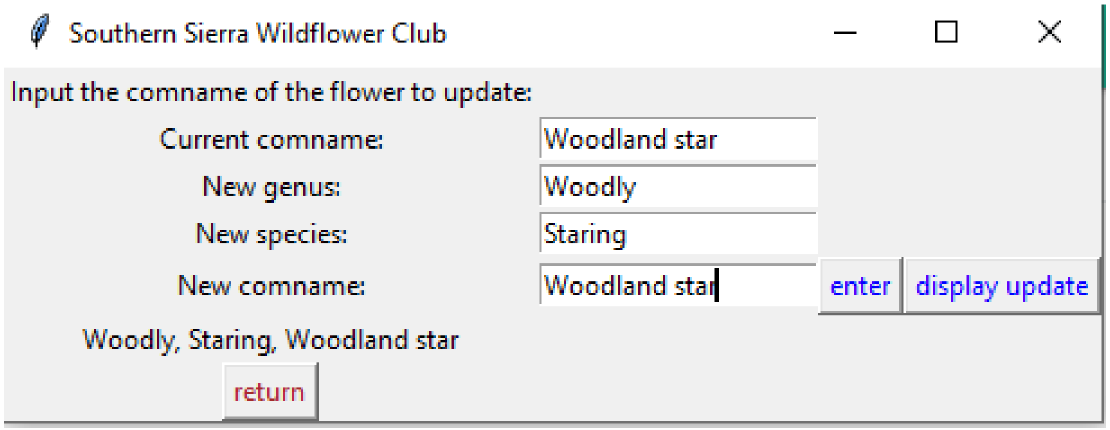


Input the comname of the flower to update:

Current comname:	Death camas
New genus:	Do not eat
New species:	Deadly
New comname:	Toxic

Do not eat, Deadly, Toxic

Image U.4: Example Death camas Result



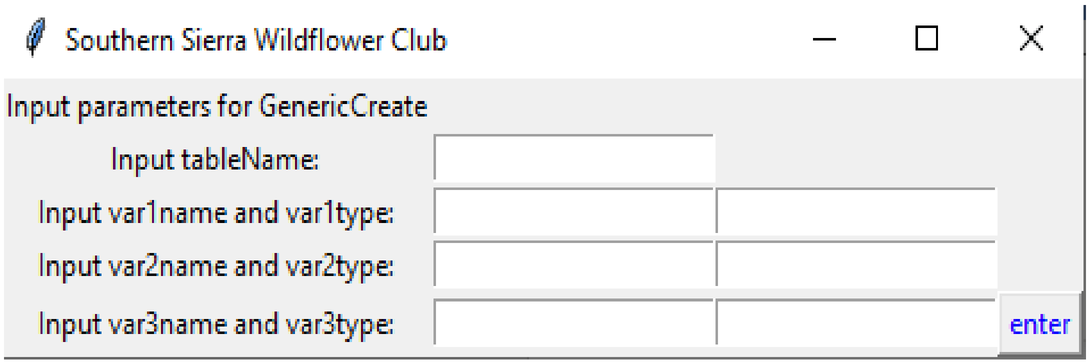
Input the comname of the flower to update:

Current comname:	Woodland star
New genus:	Woodly
New species:	Staring
New comname:	Woodland star

Woodly, Staring, Woodland star

Image U.5: Example Woodland star Result

<u>Extra Credit- Allows creation, update, and delete through a graphical interface.</u>
<u>Creation:</u>
Upon clicking the "GenericCreate" button, the user will be prompted with the screen as shown below:

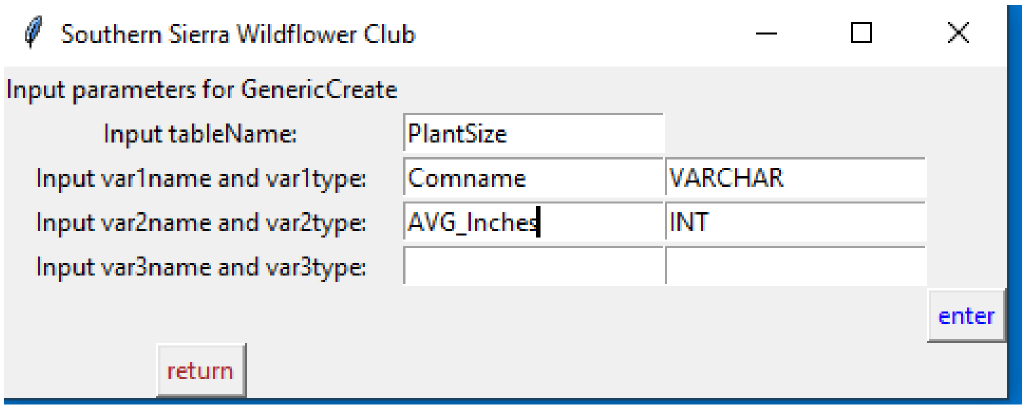

Southern Sierra Wildflower Club

Input parameters for GenericCreate

Input tableName:		
Input var1name and var1type:		
Input var2name and var2type:		
Input var3name and var3type:		

Image EC.2: GenericCreate Input Screen

To create a new table the user must input a table name and at least one variable (column) name and its data type. Upon pressing enter it will create a new table with the information in the database. This is shown below:

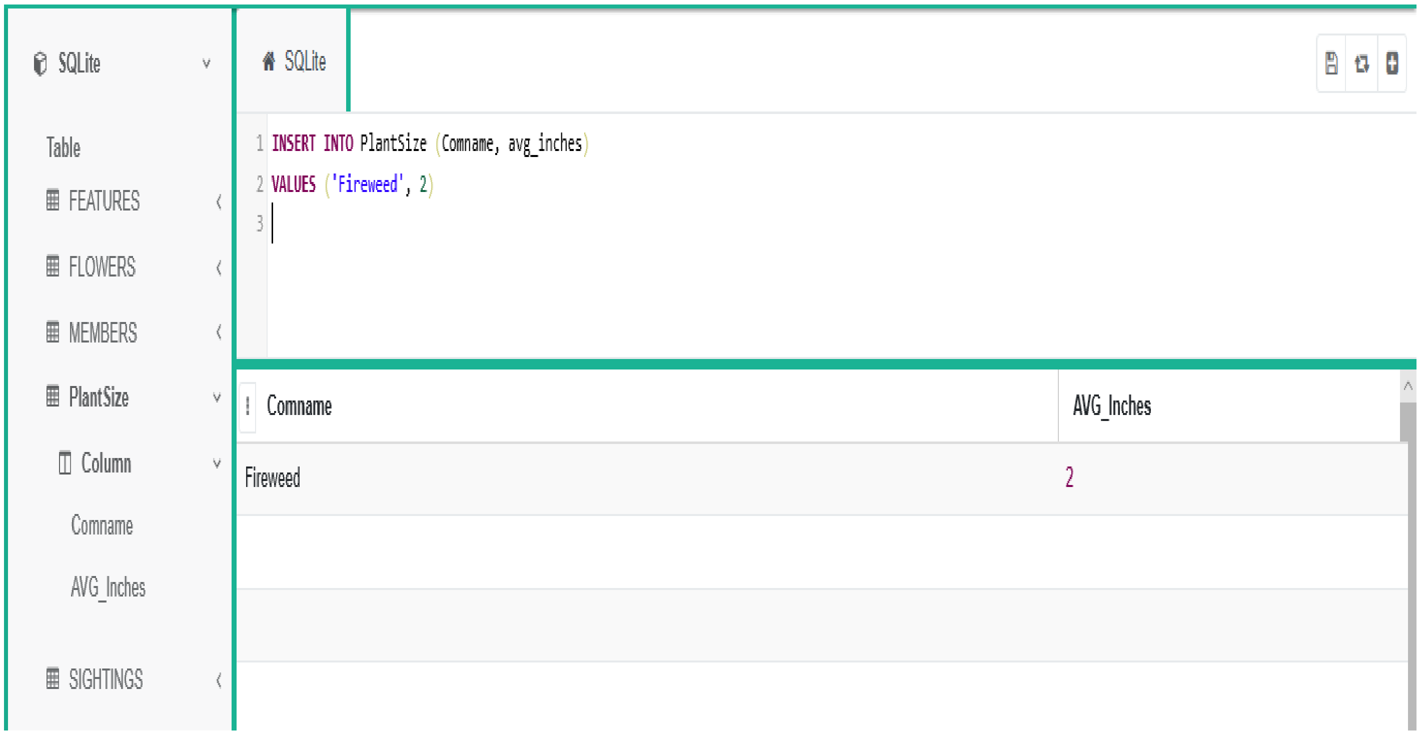

Southern Sierra Wildflower Club

Input parameters for GenericCreate

Input tableName:	PlantSize	
Input var1name and var1type:	Comname	VARCHAR
Input var2name and var2type:	AVG_Inches	INT
Input var3name and var3type:		

Image EC.3: Example PlantSize

The user can type in new information in the input boxes and press enter to create a new table or click the return button down below to go back to the Main menu. SQLite must be used to check that the new table has been created as shown below:



```
INSERT INTO PlantSize (Comname, avg_inches)
VALUES ('Fireweed', 2)
```

Comname	AVG_Inches
Fireweed	2

Image EC.4: Example PlantSize Result

Update:

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

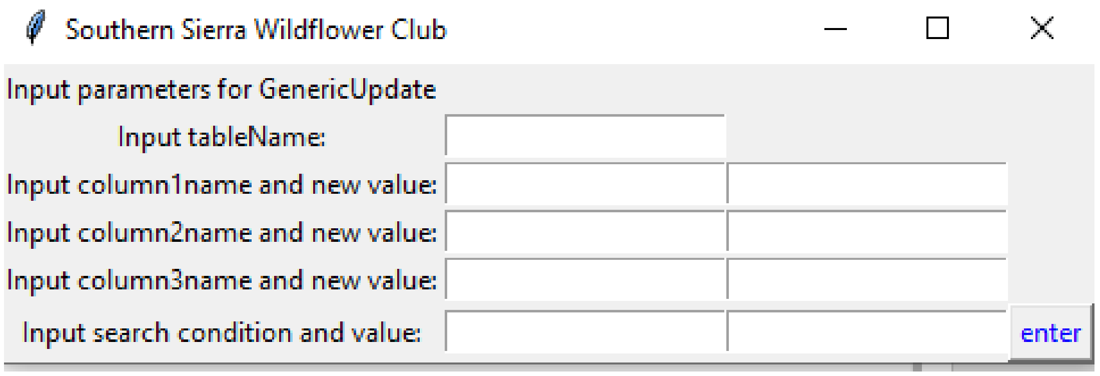
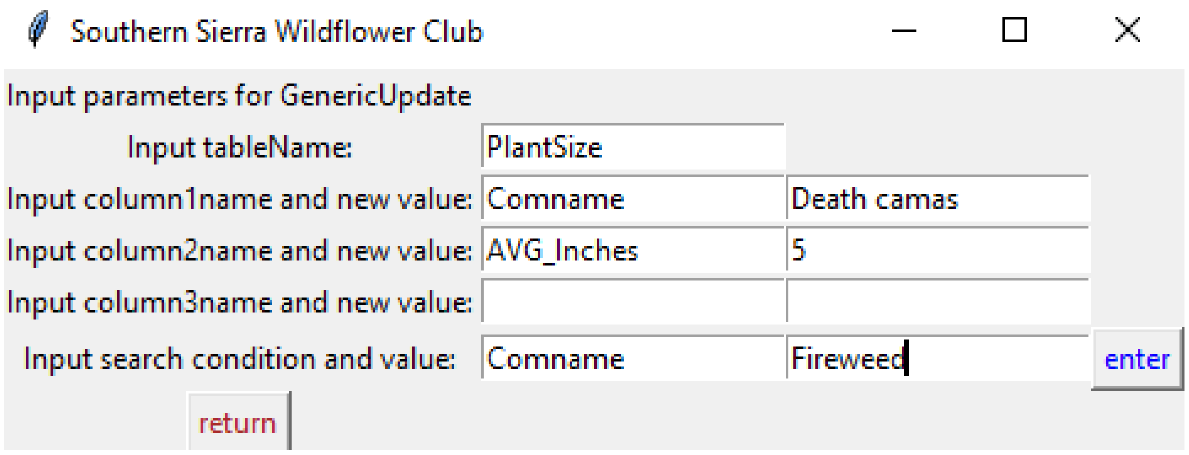

Southern Sierra Wildflower Club
Input parameters for GenericUpdate
Input tableName:
Input column1name and new value:
Input column2name and new value:
Input column3name and new value:
Input search condition and value:
enter

Image EC.5: GenericUpdate Input Screen

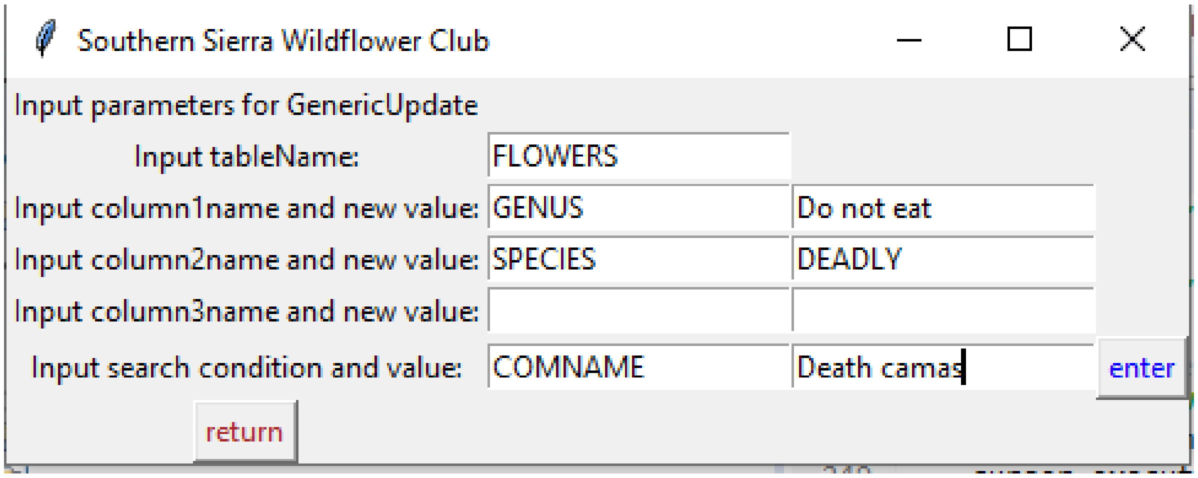
To update a table the user must input a table name, at least one column name and its new value, and the search condition. Upon pressing enter it will update the table with the information in the database. This is shown below:


Southern Sierra Wildflower Club

Input parameters for GenericUpdate

Input tableName:	PlantSize	
Input column1name and new value:	Comname	Death camas
Input column2name and new value:	AVG_Inches	5
Input column3name and new value:		
Input search condition and value:	Comname	Fireweed

Image EC.6: Example PlantSize Update

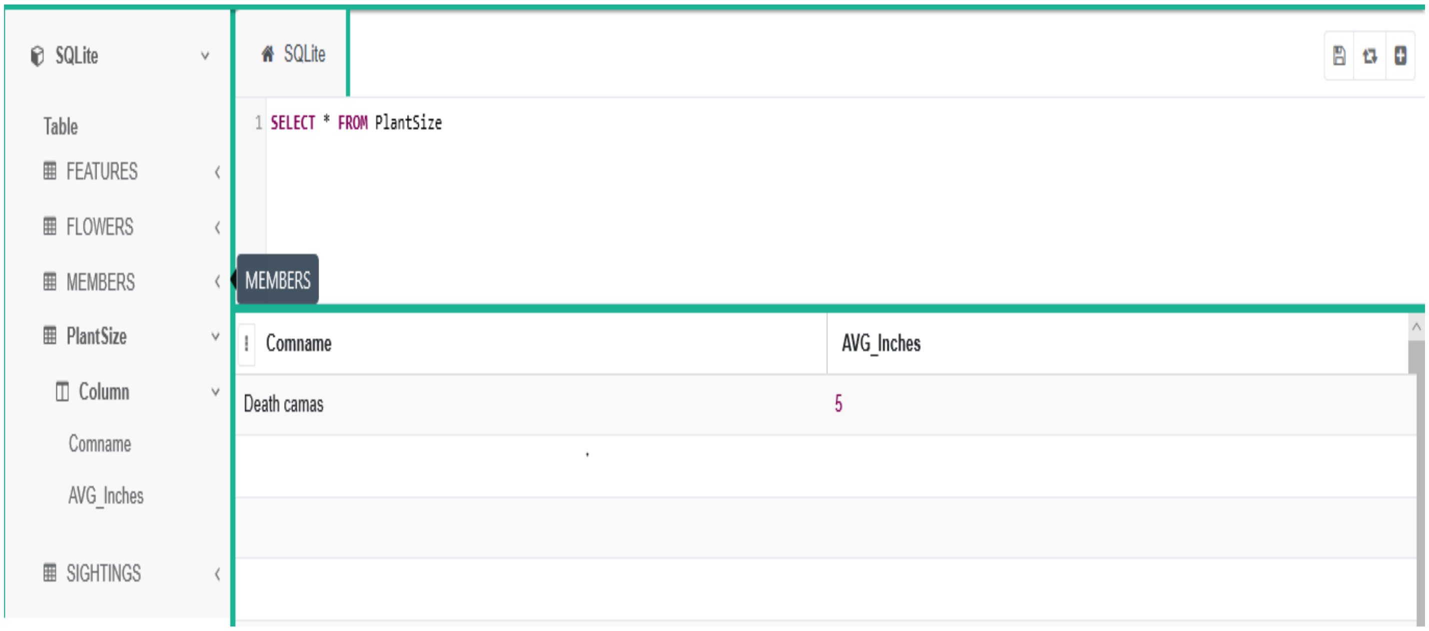

Southern Sierra Wildflower Club

Input parameters for GenericUpdate

Input tableName:	FLOWERS	
Input column1name and new value:	GENUS	Do not eat
Input column2name and new value:	SPECIES	DEADLY
Input column3name and new value:		
Input search condition and value:	COMNAME	Death camas

Image EC.7: Example FLOWERS Update

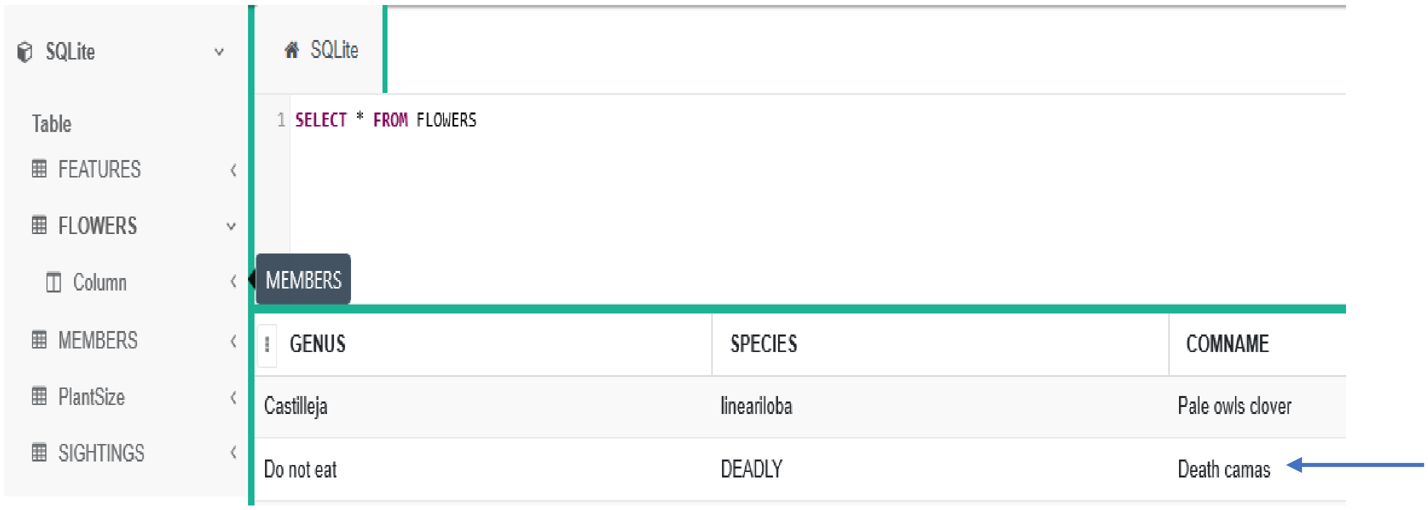
The user can type in new information in the input boxes and press enter to update a table or click the return button down below to go back to the Main menu. SQLite must be used to check that the table has been updated as shown below:



```
SELECT * FROM PlantSize
```

Comname	AVG_Inches
Death camas	5

Image EC.8 Example PlantSize Update Result



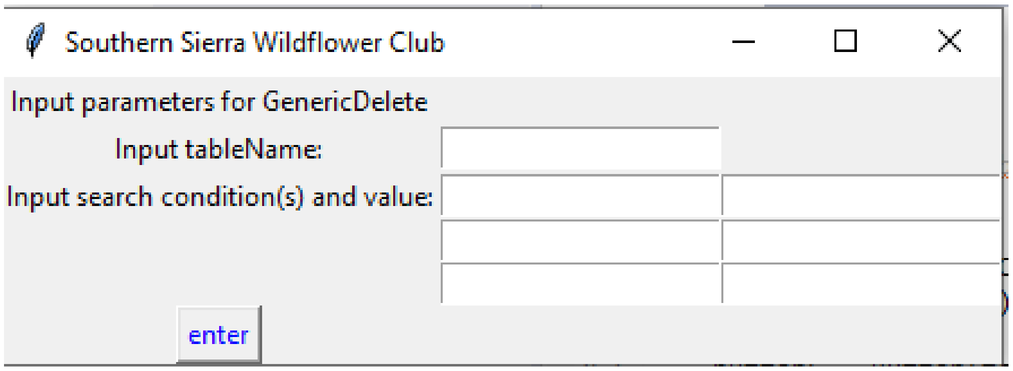
```
SELECT * FROM FLOWERS
```

GENUS	SPECIES	COMNAME
Castilleja	lineariloba	Pale owls clover
Do not eat	DEADLY	Death camas

Image EC.8 Example FLOWERS Update Result

<u>Delete:</u>

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


Southern Sierra Wildflower Club

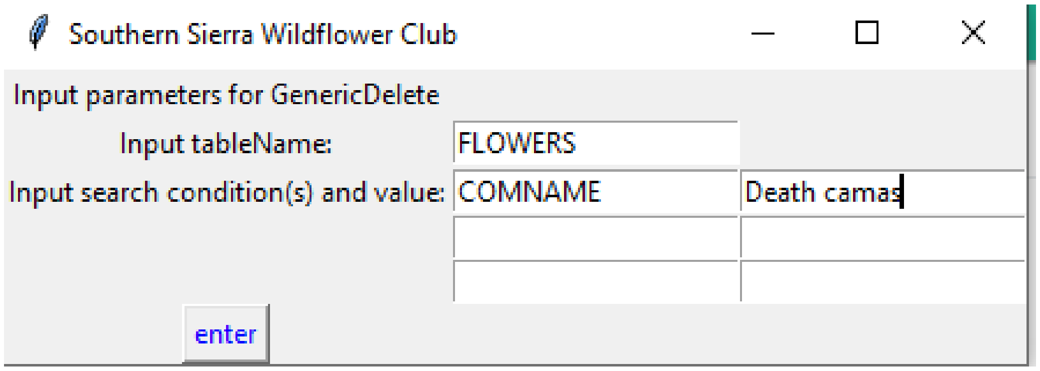
Input parameters for GenericDelete

Input tableName:

Input search condition(s) and value:

Image EC.8: GenericDelete Input Screen

To update a table the user must input a table name, at least one column name and its search condition. Upon pressing enter it will delete the row from the database. This is shown below:

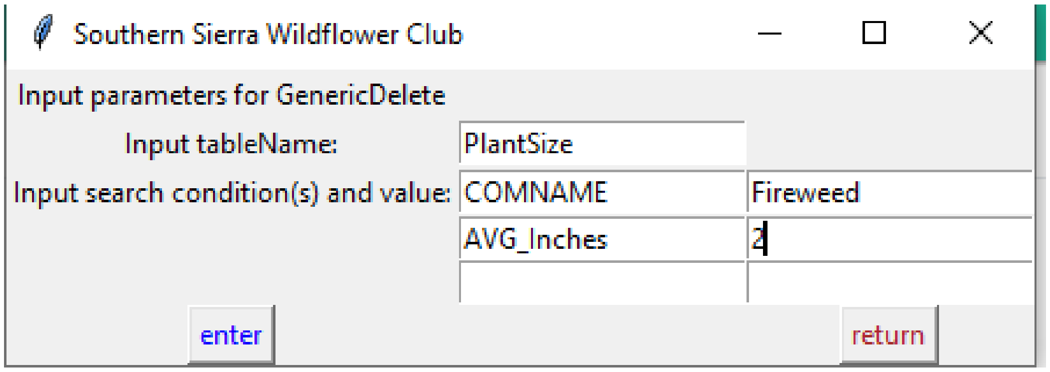

Southern Sierra Wildflower Club

Input parameters for GenericDelete

Input tableName: FLOWERS

Input search condition(s) and value: COMNAME Death camas

Image EC.9: Example FLOWERS Delete


Southern Sierra Wildflower Club

Input parameters for GenericDelete

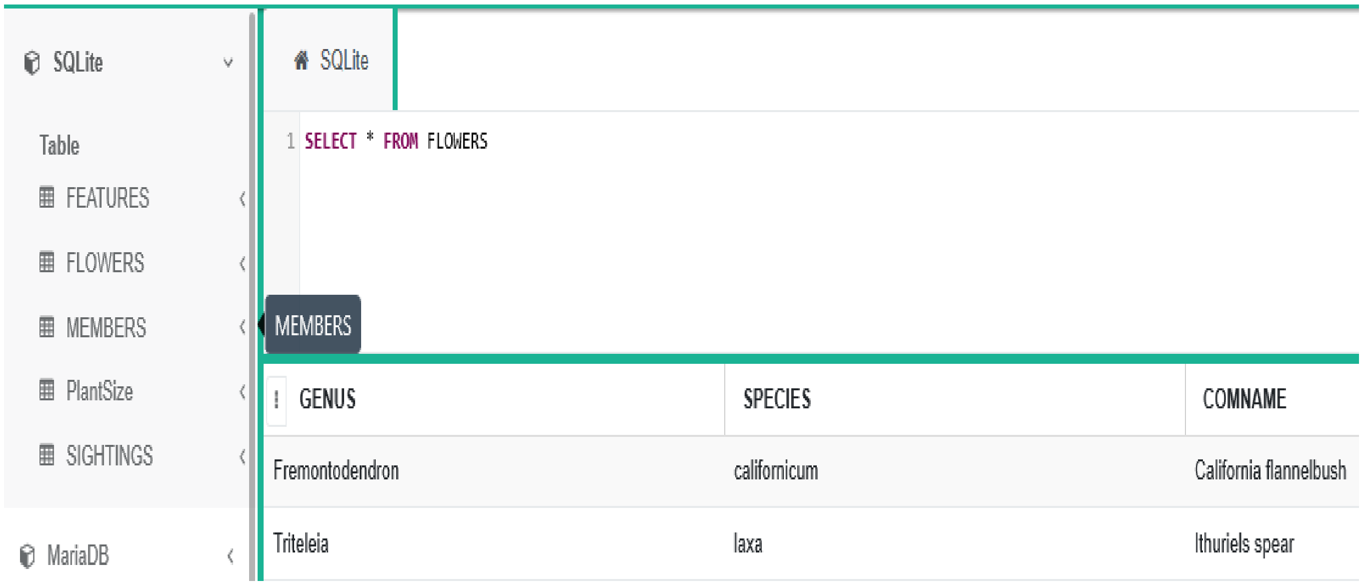
Input tableName: PlantSize

Input search condition(s) and value: COMNAME Fireweed

AVG_Inches 2

Image EC.10: Example PlantSize Delete

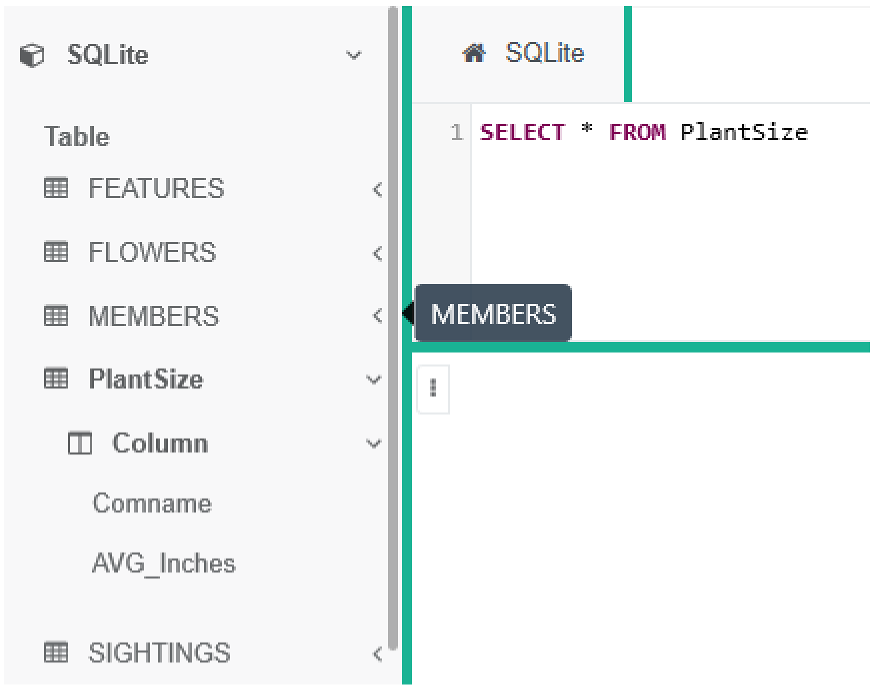
The user can type in new information in the input boxes and press enter to delete a row or click the return button down below to go back to the Main menu. SQLite must be used to check that the row has been deleted as shown below:



```
SELECT * FROM FLOWERS
```

GENUS	SPECIES	COMNAME
Fremontodendron	californicum	California flannelbush
Triteleia	laxa	Ithuriels spear

Image EC.11: Example FLOWERS Delete Result



```
SELECT * FROM PlantSize
```

Image EC.12: Example PlantSize Delete Result

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

Code- For python.

```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)
topFrame6 = Frame(root)
topFrame7 = 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 == "GenericCreate":
        topFrame5.pack()
    if s == "GenericUpdate":
        topFrame6.pack()
    if s == "GenericDelete":
        topFrame7.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):
    topFrame7.pack_forget()
    topFrame6.pack_forget()
    topFrame5.pack_forget()
    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)

#Selection: Generic Creation
def mainoption6(event):
    #Variables
    count = 0
    s=""
    s1=""
    s2=""
    s3=""
    #Format SQL
    if a7.get():
        s1+=a7.get()+" "+a8.get()
        count+=1
    if a9.get():
        s2+=a9.get()+" "+a10.get()
        count+=1
    if a11.get():
        s3+=a11.get()+" "+a12.get()
        count+=1
    if(count==3):
        s1+=",\n"
        s2+=",\n"
    if(count==2):
        s1+=",\n"
    s=s1+s2+s3
    #Create new table in db
    strCommand="CREATE TABLE " + a6.get()+ " ( "+ s + ");"
    print(strCommand);    #Testing: Print SQL
    cursor.execute(strCommand)
    connection.commit()
    #Return button
    button5 = Button(topFrame5, text="return", fg = "brown")
    button5.grid(row=8)
    button5.bind("<Button-1>", mainoption3)
```

```
#Check the string if it is a digit
def checkStr(string):
    #Format digit for SQL
    if(string.isdigit()==False):
        return ("\""+string+"\"")
    return string

#Selection: Generic Update
def mainoption7(event):
    #Variable
    count = 0
    s=""
    s1=""
    s2=""
    s3=""
    #Format SQL
    if b7.get():
        s1+=b7.get()+" = "+checkStr(b8.get())
        count+=1
    if b9.get():
        s2+=b9.get()+" = "+checkStr(b10.get())
        count+=1
    if b11.get():
        s3+=b11.get()+" = "+checkStr(b12.get())
        count+=1
    if(count==3):
        s1+=",\n"
        s2+=",\n"
    if(count==2):
        s1+=",\n"
    s=s1+s2+s3
    #Update table in db
    strCommand="UPDATE " + b6.get()+ " SET " + s+" WHERE " + b13.get() + " = " +
checkStr(b14.get())
    print(strCommand)
    cursor.execute(strCommand)
    connection.commit()
    #Return button
    button5 = Button(topFrame6, text="return", fg = "brown")
    button5.grid(row=10)
    button5.bind("<Button-1>", mainoption3)

#Selection: Generic Delete
def mainoption8(event):
    #Variable
    count = 0
    s=""
    s1=""
    s2=""
    s3=""
    #Format SQL
    if e7.get():
        s1+=e7.get()+" = "+checkStr(e8.get())
        count+=1
    if e9.get():
        s2+=e9.get()+" = "+checkStr(e10.get())
        count+=1
    if e11.get():
        s3+=e11.get()+" = "+checkStr(e12.get())
        count+=1
    if(count==3):
        s1+="OR \n"
        s2+="OR \n"
    if(count==2):
        s1+="OR \n"
    s=s1+s2+s3
    #Delete table in db
    strCommand="DELETE FROM " + e6.get()+ " WHERE " + s
    print(strCommand)   #Testing: Print SQL
    cursor.execute(strCommand)
    connection.commit()
    #Return button
    button5 = Button(topFrame7, text="return", fg = "brown")
    button5.grid(row=5, column=2)
    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 )
button4 = Button(topFrame,  text="GenericCreate", fg = "purple")
button4.bind("<Button-1>", mainoption)
button4.pack(side = LEFT )
button5 = Button(topFrame,  text="GenericUpdate", fg = "purple")
button5.bind("<Button-1>", mainoption)
button5.pack(side = LEFT )
button6 = Button(topFrame,  text="GenericDelete", fg = "purple")
button6.bind("<Button-1>", mainoption)
button6.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)

#Generic Create menu text coloring
theLabel = Label(topFrame5, text ="Input parameters for GenericCreate").grid(row=0)
Label(topFrame5, text='Input tableName:').grid(row=1)
a6 = Entry(topFrame5)
```

```
a6.grid(row=1, column=1)
Label(topFrame5, text='Input var1name and var1type:').grid(row=2)
a7 = Entry(topFrame5)
a7.grid(row=2, column=1)
a8 = Entry(topFrame5)
a8.grid(row=2, column=2)
Label(topFrame5, text='Input var2name and var2type:').grid(row=3)
a9 = Entry(topFrame5)
a9.grid(row=3, column=1)
a10 = Entry(topFrame5)
a10.grid(row=3, column=2)
Label(topFrame5, text='Input var3name and var3type:').grid(row=4)
a11 = Entry(topFrame5)
a11.grid(row=4, column=1)
a12 = Entry(topFrame5)
a12.grid(row=4, column=2)
button6 = Button(topFrame5, text="enter", fg = "blue")
button6.grid(row=5, column = 3)
button6.bind("<Button-1>", mainoption6)

#Generic Update menu text coloring
theLabel = Label(topFrame6, text ="Input parameters for GenericUpdate").grid(row=0)
Label(topFrame6, text='Input tableName:').grid(row=1)
b6 = Entry(topFrame6)
b6.grid(row=1, column=1)
Label(topFrame6, text='Input column1name and new value:').grid(row=3)
b7 = Entry(topFrame6)
b7.grid(row=3, column=1)
b8 = Entry(topFrame6)
b8.grid(row=3, column=2)
Label(topFrame6, text='Input column2name and new value:').grid(row=5)
b9 = Entry(topFrame6)
b9.grid(row=5, column=1)
b10 = Entry(topFrame6)
b10.grid(row=5, column=2)
Label(topFrame6, text='Input column3name and new value:').grid(row=7)
b11 = Entry(topFrame6)
b11.grid(row=7, column=1)
b12 = Entry(topFrame6)
b12.grid(row=7, column=2)
Label(topFrame6, text='Input search condition and value:').grid(row=9)
b13 = Entry(topFrame6)
b13.grid(row=9, column=1)
b14 = Entry(topFrame6)
b14.grid(row=9, column=2)
button6 = Button(topFrame6, text="enter", fg = "blue")
button6.grid(row=9, column = 3)
button6.bind("<Button-1>", mainoption7)

#Generic Delete menu text coloring
theLabel = Label(topFrame7, text ="Input parameters for GenericDelete").grid(row=0)
Label(topFrame7, text='Input tableName:').grid(row=1)
e6 = Entry(topFrame7)
e6.grid(row=1, column=1)
Label(topFrame7, text='Input search condition(s) and value:').grid(row=2)
```

```
e7 = Entry(topFrame7)
e7.grid(row=2, column=1)
e8 = Entry(topFrame7)
e8.grid(row=2, column=2)
e9 = Entry(topFrame7)
e9.grid(row=3, column=1)
e10 = Entry(topFrame7)
e10.grid(row=3, column=2)
e11 = Entry(topFrame7)
e11.grid(row=4, column=1)
e12 = Entry(topFrame7)
e12.grid(row=4, column=2)
button6 = Button(topFrame7, text="enter", fg = "blue")
button6.grid(row=5)
button6.bind("<Button-1>", mainoption8)

root.mainloop() #Infinite loop
```