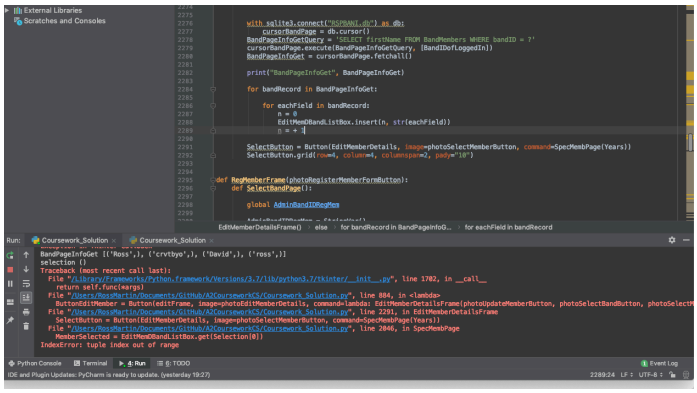
6 Developmental Testing

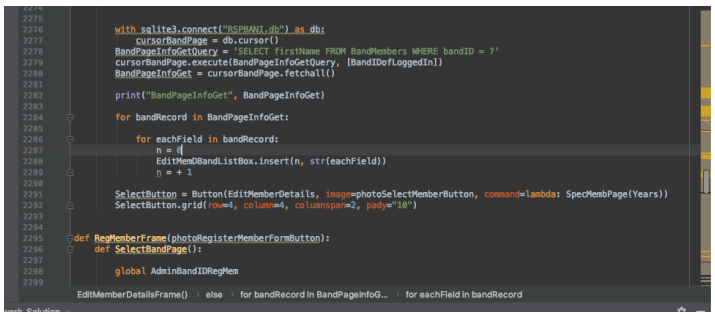
6.1 Developmental testing; Produce evidence of testing at each stage of development.

In the early stages of my development, when I was trying to design the layout of my program. It took me a wee while as I had a few teething problems, I had never used the grid() method for placing objects into a frame. Once I thought I had mastered it, I had programmed the left button menu bar to be a fixed size but for some reason, it shrank to the size of the largest button which left the root (purple) of my program visible as my frames were not covering the whole screen. I was able to find my solution using stack overfull for some guidance. I had to configure the column that the left menu bar was in to a minimum size – so I set the min size to the size I wanted it to be and that solved my issue.

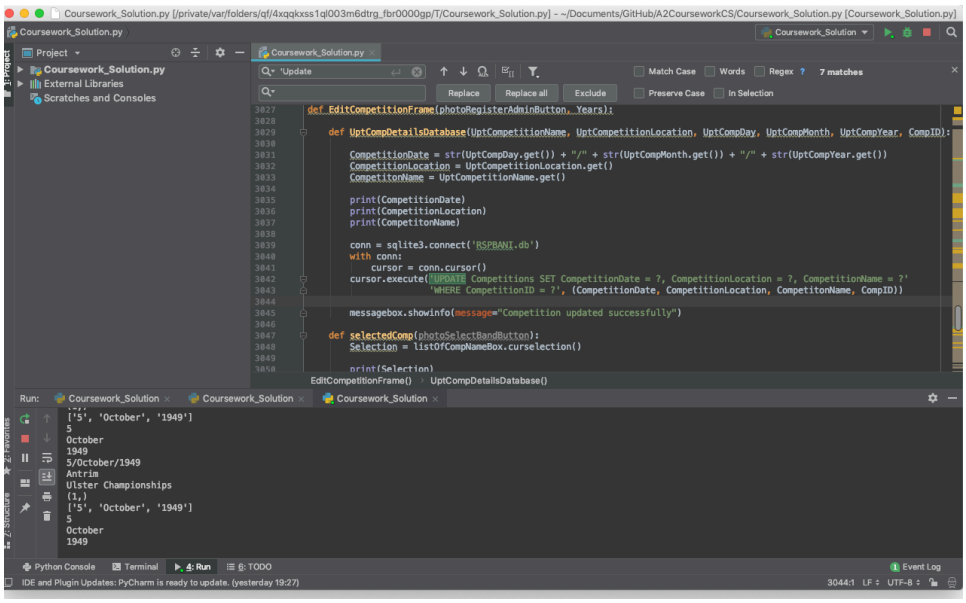
I ran into a problem where, I was creating a list box and populating it with all the members of a band. The user should then be able to select a member and press the Edit Member Details button which just called a function and passed the list box as an argument of the function. However, when I tried to do this my system crashed and I got the following error message as my program failed to display the list box and give the user the option to select their member as it had continued on with the function called by the button before the button had actually been pressed.

After some research online, I was able to find the reasoning behind this issue. When I was stating the function to be called on click of



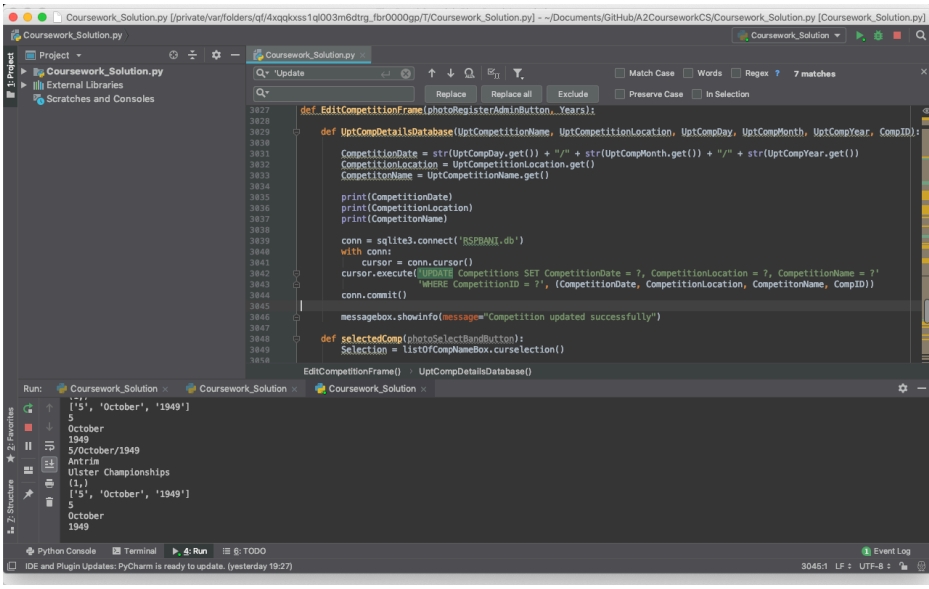


the button, the command function cannot accept arguments along with the function or procedure that you want to call. To fix this problem, before I called the function name with arguments I had to write “command=lambda: function(arguments)” instead of “command=function(arguments)” in front and this tells the command argument inside the button that parameters are to be passed through this function.



When I began to save data my database that the user had inputted, I ran into another issue. I had been able to get all the data the user had entered into the entry boxes on the form. However, when I was saving the data to the database once I pressed the save button and checked the database to ensure it then existed in the database it didn't save to the database and I wasn't getting an error message, so I found it very difficult to resolve the issue. I got one of my peers to have a look at my code to see if they would see where I was going wrong as the code was still running so I knew it wasn't a syntax error instead a logical error when saving to the database. This was my code beforehand with no error message and no data saved to the database:

They were able to fix where I had gone wrong. In fact, they spotted it straight away. I had forgotten to commit my change to the database hence why it wasn't saving. I had to add one line of code "conn.commit()"



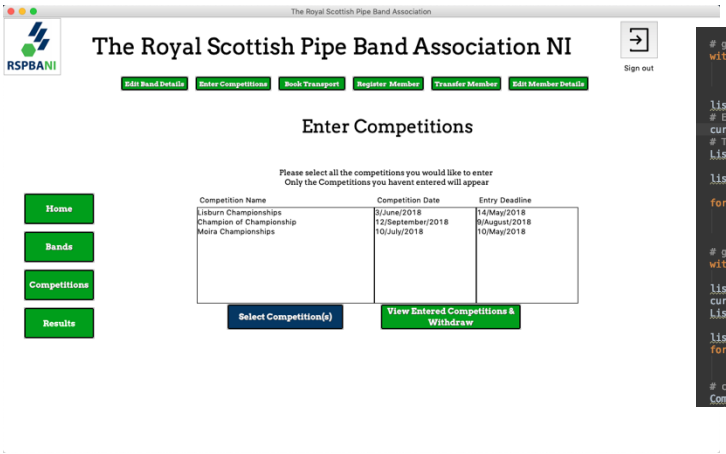
I also came across a problem trying to use while loops with the time.sleep function in Tkinter. This issue occurred with me when I was trying to implement my photo gallery. I was trying to get my homepage to change photo every 10 seconds and the only way I could figure out how to implement this was using the sleep function. I ran the program, and everything worked nicely until I tried to navigate to another page, and this is where the problems began. As I was sleeping my application when I tried to change page there would be a delay in the action due to my program "sleeping". This was a big performance issue as I could not have my program looking like it wasn't responding to user driven event clicks. Here is the code for

what I tried to achieve unsuccessfully. In the end I had to take out this feature as my system performance was more important than my photo gallery even though it would have been nice to have.

```
                                "association in Northern Ireland and ",
                        foreground="black", font=("Arvo"))
    lblAboutUs2.grid(row=4, column=0, columnspan=9, sticky="w")

    lblAboutUs3 = Label(Home,
                        text="over three thousand individual members.",
                        foreground="black", font=("Arvo"))
    lblAboutUs3.grid(row=5, column=0, columnspan=9, sticky="w")

    # create the canvas, size in pixels
    canvas = Canvas(Home, width=600, height=400)

    # pack the canvas into a frame/form
    canvas.grid(row=6, column=0, columnspan=5)

    while True:
        for i in range(1, 12):
            print(i)
            photo = os.path.join('Photos', str(i) +'.png')
            # load the .gif image file
            global photoimage
            photoimage = PhotoImage(file=photo)
            # put gif image on canvas
            # pic's upper left corner (NW) on the canvas is at x=50 y=10
            canvas.create_image(50, 10, image=photoimage, anchor=NW)
            root.update()
            time.sleep(10)

        # Add content to home page
        # Photo Gallery
        # Latest Results - Include competition date, location, and first place in each grade
```

As I was validating my “Judging Competitions” feature I had another problem to solve. To ensure that the judges could only enter valid scores for the bands in each grade. I had hoped to create a list of the possible scores that the judges could select from using the drop list widget of Tkinter. The issue revolved around the fact that I didn’t know exactly how many bands had entered the competition so the number of drop lists I needed to display was also unknown. This made this very hard as I had to populate my frame with a certain number of these widgets. I was able to figure out how to use a for loop in the range of the number of bands, but the issue came when I tried to get the values that the judge had selected for each band. This was because I could not dynamically assign the textvar variables used to store the chosen number for the unknown number of drop lists. In the end I could not use this method and I had to revert back to entry boxes and the traditional validation. Unfortunately, I did not take a screenshot of this however, at one stage trying to debug this issue I did ask my teacher, of which she witnessed my issue, if she could provide me with any ideas but between ourselves, we couldn’t find a possible solution to my problem.

I had an issue where I thought I was only inserting the competitions that a band hadn’t already entered into the list box on the “Enter competitions” page, however I realised that this logic was incorrect. I had to populate the list box with competitions that the band hadn’t entered, and of which hadn’t been drawn.
This was the list of competitions that I thought the band hadn’t entered and still could. I was getting a list of all the competitions and a list of all the competitions the band had entered and returning a list of the difference.




```python
# gets a list of all the competitions that have not yet been drawn and is still possible to enter
with sqlite3.connect("RSPBANI.db") as db:
    # Creates a cursor to search through the data
    cursorListOfComps = db.cursor()
    # SQL Query to select all Competitions from the competitions table
listOfCompsQuery = 'SELECT CompetitionID FROM Competitions'
# Execute the query to get a list of all the bands
cursorListOfComps.execute(listOfCompsQuery)
# The query will produce a 2 dimensional tuple
ListOfComps = cursorListOfComps.fetchall()

listOfComps = []

for i in ListOfComps:
    for j in i:
        listOfComps.append(str(j))

# get a list of the Competitions the Band has already entered
with sqlite3.connect("RSPBANI.db") as db:
    cursor = db.cursor()
listOfCompsEnteredQuery = 'SELECT CompetitionID FROM BandsEntered WHERE BandID = ?'
cursor.execute(listOfCompsEnteredQuery, [BandIDComps])
ListOfCompsEntered = cursor.fetchall()

listOfCompsEntered = []
for i in ListOfCompsEntered:
    for j in i:
        listOfCompsEntered.append(j)
# compare these two lists and return the competitions that havent been entered and can still be entered
CompsNotEntered = diff(listOfCompsEntered, listOfComps)
```

Instead to ensure a band could not enter a competition that had been already drawn.
To only get the competitions that hadn't be drawn and not entered by the band I had to query the database again to get a list of all the competitions that hadn't been drawn (Drawn = "False") and store the result in a list. I then had to increment through this list and for each element in the list try and remove it from the list of competitions not entered by the band, if it existed using the try except clause. This then gave me a list of all the competitions the band hadn't entered which haven't been drawn.

```
Replace  Replace all  Exclude  Preserve Case  In Selection

        with sqlite3.connect("RSPBANI.db") as db:
            # Creates a cursor to search through the data
            cursorListOfComps = db.cursor()
            # SQL Query to select all Competitions from the competitions table
        listOfCompsQuery = 'SELECT CompetitionID FROM Competitions WHERE Drawn = ?'
        # Execute the query to get a list of all the bands
        cursorListOfComps.execute(listOfCompsQuery, ["True"])
        # The query will produce a 2 dimensional tuple
        ListOfCompTrue = cursorListOfComps.fetchall()

        ListCompsAlreadyDrawn = []

        for i in ListOfCompTrue:
            for j in i:
                ListCompsAlreadyDrawn.append(str(j))

        # get a list of the Competitions the Band has already entered
        with sqlite3.connect("RSPBANI.db") as db:
            cursor = db.cursor()
        listOfCompsEnteredQuery = 'SELECT CompetitionID FROM BandsEntered WHERE BandID = ?'
        cursor.execute(listOfCompsEnteredQuery, [BandIDComps])
        ListOfCompsEntered = cursor.fetchall()

        listOfCompsEntered = []
        for i in ListOfCompsEntered:
            for j in i:
                listOfCompsEntered.append(str(j))
                print(listOfCompsEntered)

        # compare these two lists and return the competitions that havent been entered. This list s
        CompsNotEntered = diff(listOfCompsEntered, listOfComps)
        # initalise a dictionary to store the competition name and date from all the competitions t
        # still be entered

        for element in ListCompsAlreadyDrawn:
            try:
                CompsNotEntered.remove(str(element))
            except:
                None

        CompNameDateNotEntered = {}
```

When I tried to import the python docx library, so I could use it to produce my outputs from the system. I got this error saying that no module was found. I googled the solution and I was told that I need to install the module onto my

```
Traceback (most recent call last):
  File "D:\Testing_A2_Coursework.py", line 8, in <module>
    from docx import Document
ImportError: No module named 'docx'
```

mac. I opened terminal and ran the pip install python-docx command. I however got another error, this time I didn't have pip installed. I then had to download the pip file and run it on my mac before I could import the python docx module.