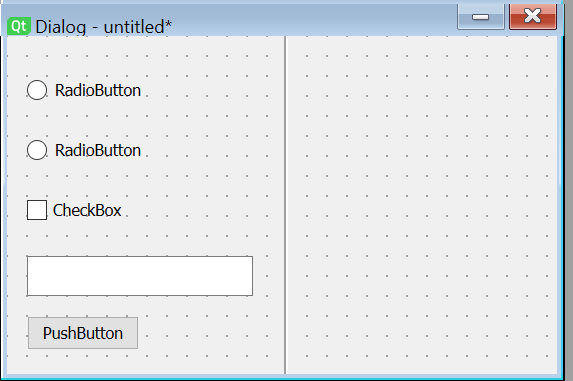
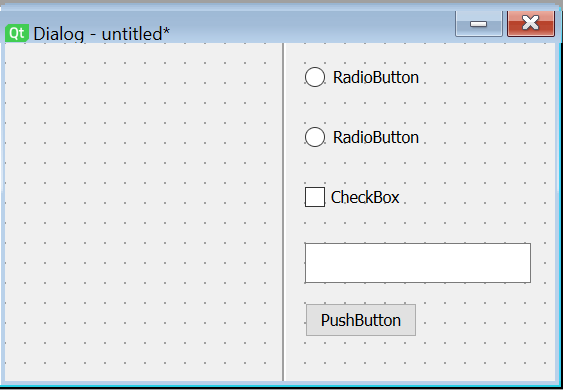
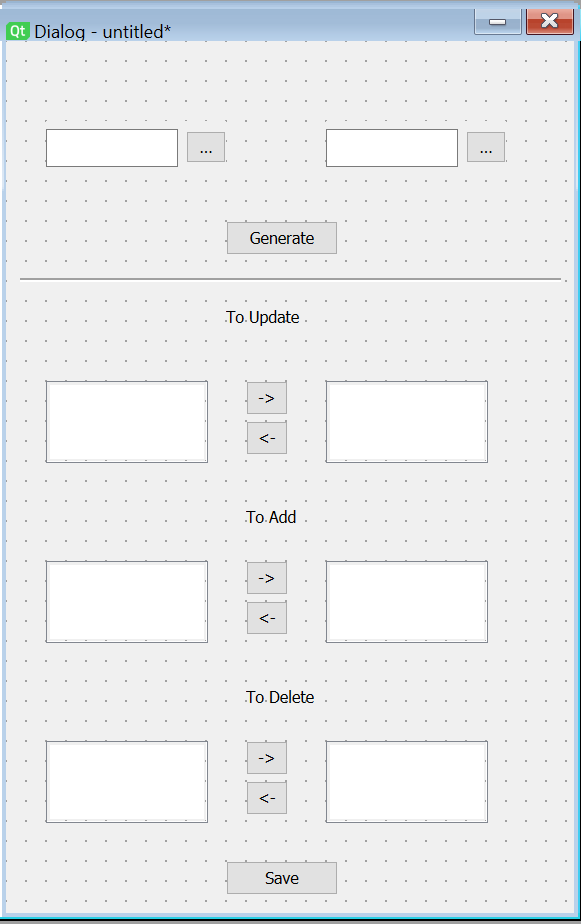
**UI UPDATE TOOL**

User has a python file contains code for a UI that looks like this



User has a .ui file that looks like this and wants their python file to also look like this





**Part 1 -> runs when generate is pressed**

GET FILES

def(getUIFiles): #Get the location of the .ui file  
 *'''  
 First get the location of the ui by capturing the "text" inside the Q file widget  
 Return an error if you can't find the file it  
 kick to exportUItoPY  
 '''*def(exportUItoPY): # HARD!! Save the .ui as .py  
 *'''  
 Use the pyuic found inside PyQt5 to convert the file  
 and save it inside the plugin folder  
 save the lines in the .py file as "genpylines"  
 kick to getPyFile  
 '''*def(getPyFile): # Open and save whats inside the .py file  
 *'''  
 Use "with" to open python file and save the lines to a list  
 Each line should be a new value in the list so use a for loop and .append  
 Save this as "userpylines"  
 '''*

Parse Files (x2 one for genpy and one for userpy)

def(parseRadioButton): # parse genpylines for all radiobuttons and save values  
 *'''  
 Go through pylines and extract lines that contain the radiobutton class  
 capture each different radiobutton as a new dictionary entry with the qt name as the key value  
 Find a way to save the important values under the dictionary  
 possible ways:  
 use the len() of key value and the known length of the rest to index the values out  
 If always at the end use [-index] for just the values we want <-The best if this is ture  
 do a search for the stuff just before what we want an make a index of that  
 Save values as radiovalues  
 '''*def(parseTextBoxes): # parse genpylines for all textboxes and save values  
  
  
def(parseCheckBoxes): # parse genpylines for all checkboxes and save values  
  
def(parsePushButtons): # parse genpylines for all pushButtons and save values

Generate Lists

def(generateChangeList) # Find keynames have different values in genpy and userpy  
  
def(generateAddList) # Find keywords in genpy that aren't in userpy  
  
def(generateDeleteList) # find keywords that are in userpy but not genpy

**Part 2 -> runs when save is pressed**

Update lists

def(updateLists) # Update each list based on user input

Update PyFile

def(toupdateRadioButton): # Look for radiobuttons in userpylines that need updating and update them  
 *'''  
 update the lines with the values saved in radiovalues  
 this should be a for loop so it does this for each keyvalue  
 write the values back into the original user .py file  
 '''*def(updateCheckBoxes): # Look for check boxes in userpylines that need updating and update them  
  
def(updateTextBoxes): # Look for text boxes in userpylines that need updating and update them  
  
def(updatePushButtons): #Look for pushbuttons in userpylines that need updating and update them