import datetime  
import time  
from PyQt5 import uic  
import sqlite3  
from PyQt5 import QtWidgets  
from PyQt5.QtWidgets import QApplication, QMessageBox, QTableWidgetItem, QLabel, QWidget, QBoxLayout  
from PyQt5.QtCore import Qt  
from PyQt5.QtGui import QColor  
  
Form, Window = uic.loadUiType("mainwindow.ui")  
app = QApplication([])  
window = Window()  
form = Form()  
form.setupUi(window)  
  
window.setWindowTitle("Diplom")  
window.resize(241, 221)  
form.frame\_2.hide()  
form.frame\_3.hide()  
form.perehod.hide()  
conn = sqlite3.connect("database.db")  
cur = conn.cursor()  
cur.execute('CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY AUTOINCREMENT, login TEXT, paswd TEXT)')  
conn.commit()  
cur.execute('CREATE TABLE IF NOT EXISTS reb (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT)')  
conn.commit()  
cur.execute('CREATE TABLE IF NOT EXISTS antena (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT)')  
conn.commit()  
cur.execute('CREATE TABLE IF NOT EXISTS note (id INTEGER PRIMARY KEY AUTOINCREMENT, login TEXT, star TEXT, stopp TEXT)')  
conn.commit()  
form.label\_9.hide()  
def CHECK():  
 if form.checkBox.isChecked():  
 if form.checkBox\_2.isChecked():  
 if form.checkBox\_3.isChecked():  
 if form.checkBox\_4.isChecked():  
 if form.checkBox\_5.isChecked():  
 if form.checkBox\_6.isChecked():  
 if form.checkBox\_7.isChecked():  
 form.label\_8.setText("Инструкция полета:")  
 form.checkBox.hide()  
 form.checkBox\_2.hide()  
 form.checkBox\_3.hide()  
 form.checkBox\_4.hide()  
 form.checkBox\_5.hide()  
 form.checkBox\_6.hide()  
 form.checkBox\_7.hide()  
 form.label\_9.show()  
 form.perehod.show()  
  
  
form.checkBox.stateChanged.connect(CHECK)  
form.checkBox\_2.stateChanged.connect(CHECK)  
form.checkBox\_3.stateChanged.connect(CHECK)  
form.checkBox\_4.stateChanged.connect(CHECK)  
form.checkBox\_5.stateChanged.connect(CHECK)  
form.checkBox\_6.stateChanged.connect(CHECK)  
form.checkBox\_7.stateChanged.connect(CHECK)  
  
  
  
  
def LOGIN():  
 user = cur.execute('SELECT login FROM users').fetchall()  
 for i in user:  
 if i == (form.login.text(),):  
 pas = cur.execute('SELECT paswd FROM users WHERE login =?', (i)).fetchall()  
 if pas[0] == (form.password.text(),):  
 form.frame.hide()  
 window.resize(661, 371)  
 form.frame\_2.move(0, 0)  
 form.frame\_2.show()  
 else:  
 msg = QMessageBox()  
 msg.setWindowTitle("Ошибка входа")  
 msg.setText("Неверный пароль")  
 msg.setIcon(QMessageBox.Warning)  
 msg.exec\_()  
  
form.vhod.clicked.connect(LOGIN)  
  
def OBNOVA\_DATA():  
 table\_1 = cur.execute("SELECT name FROM sqlite\_master WHERE type='table';").fetchall()  
 reb = cur.execute("SELECT name FROM reb").fetchall()  
 AllItems = [(form.comboBox.itemText(i),) for i in range(form.comboBox.count())]  
 AllItems\_2 = [(form.comboBox\_2.itemText(i),) for i in range(form.comboBox.count())]  
 form.progressBar\_2.setValue(0)  
 for i in table\_1:  
 a = ('"' + str(i[0]) + '"',)  
 if (a in reb) and (i not in AllItems):  
 form.comboBox.addItem(str(i[0]))  
 antena = cur.execute("SELECT name FROM antena").fetchall()  
 for i in table\_1:  
 a = ('"' + str(i[0]) + '"',)  
 if (a in antena) and (i not in AllItems\_2):  
 form.comboBox\_2.addItem(str(i[0]))  
  
def PEREHOD():  
 form.tableWidget.setHorizontalHeaderLabels(['Диапазон, МГц', 'Выбор антенны'])  
 form.frame\_2.hide()  
 form.frame\_3.move(0, 0)  
 form.frame\_3.show()  
 window.resize(651, 451)  
 now = datetime.datetime.now()  
 form.label\_6.setText(str(now)[:10])  
 form.comboBox.addItem(str("Выберите комплекс"))  
 form.comboBox.setCurrentIndex(0)  
 form.tableWidget.setStyleSheet('color: rgb(0, 0, 0); background-color: rgb(206, 206, 206);')  
 OBNOVA\_DATA()  
 def text\_changed(f):  
 form.tableWidget.setRowCount(0)  
 form.tableWidget.clear()  
 form.tableWidget.setStyleSheet('color: rgb(0, 0, 0); background-color: rgb(206, 206, 206);')  
 rowPosition = form.tableWidget.rowCount()  
 index = form.comboBox.findText(str("Выберите комплекс")) # find the index of text  
 form.comboBox.removeItem(index)  
 form.tableWidget.setHorizontalHeaderLabels(['Диапазон, МГц', 'Выбор антенны'])  
 f = '"' + f + '"'  
 d = cur.execute("SELECT \* FROM " + str(f)).fetchall()  
 for i in range(len(d)):  
 form.tableWidget.insertRow(rowPosition)  
 table\_1 = cur.execute("SELECT name FROM sqlite\_master WHERE type='table';").fetchall()  
 antena = cur.execute("SELECT name FROM antena").fetchall()  
 for i in range(len(d)):  
 form.tableWidget.setItem(i, 0, QTableWidgetItem(str(d[i][0])))  
 for i in range(len(d)):  
 form.frame\_3.comboAntena = QtWidgets.QComboBox(window)  
 form.frame\_3.comboAntena.addItem("Штатная антенна")  
 for j in table\_1:  
 a = ('"' + str(j[0]) + '"',)  
 AllItems = [(form.frame\_3.comboAntena.itemText(i),) for i in range(form.comboBox.count())]  
 if (a in antena) and (j not in AllItems):  
 diap\_1 = form.tableWidget.item(i, 0).text()  
 Min, Max = map(int, str(diap\_1).split("-"))  
 diap = cur.execute('SELECT \* FROM ' + (str('"' + str(j[0]) + '"'))).fetchall()  
 Min\_2, Max\_2 = map(int, str(diap[0][0]).split('-'))  
 if Min >= Min\_2 and Max <= Max\_2:  
 form.frame\_3.comboAntena.addItem(str(j[0]))  
 form.tableWidget.setCellWidget(i, 1, form.frame\_3.comboAntena)  
 form.comboBox.currentTextChanged.connect(text\_changed)  
  
form.perehod.clicked.connect(PEREHOD)  
  
ust = 0  
  
  
  
  
#Виджет добавления  
  
widget\_dobav = QWidget()  
widget\_dobav.resize(480, 450)  
widget\_dobav.setWindowTitle("Добавление устройства")  
widget\_dobav.setStyleSheet('background-color: rgb(70, 70, 70); color: rgb(255, 255, 255); font: 12pt "MS Shell Dlg 2";')  
  
label\_dobav = QtWidgets.QLabel(widget\_dobav)  
label\_dobav.setGeometry(10, 0, 500, 30)  
  
label\_name = QtWidgets.QLabel(widget\_dobav)  
label\_name.setGeometry(10, 40, 180, 30)  
label\_name.setText("Введите название:")  
label\_name = QtWidgets.QLabel(widget\_dobav)  
label\_name.setGeometry(10, 80, 225, 30)  
label\_name.setText("Количество диапазонов:")  
  
lineName = QtWidgets.QLineEdit(widget\_dobav)  
lineName.setGeometry(201, 43, 180, 25)  
lineName.setStyleSheet("border-color: white; border-style: solid; border-width: 1px;")  
  
lineDiap = QtWidgets.QLineEdit(widget\_dobav)  
lineDiap.setGeometry(241, 83, 180, 25)  
lineDiap.setStyleSheet("border-color: white; border-style: solid; border-width: 1px;")  
  
pushDobav = QtWidgets.QPushButton(widget\_dobav)  
pushDobav.setGeometry(9, 118, 210, 25)  
pushDobav.setStyleSheet("border-color: white; border-style: solid; border-width: 1px;")  
pushDobav.setText("Добавить")  
  
pushOtobraz = QtWidgets.QPushButton(widget\_dobav)  
pushOtobraz.setGeometry(225, 118, 210, 25)  
pushOtobraz.setStyleSheet("border-color: white; border-style: solid; border-width: 1px;")  
pushOtobraz.setText("Отобразить")  
  
pushDel = QtWidgets.QPushButton(widget\_dobav)  
pushDel.setGeometry(9, 148, 430, 25)  
pushDel.setStyleSheet("border-color: white; border-style: solid; border-width: 1px;")  
pushDel.setText("Удалить")  
  
zurnal = QtWidgets.QLabel(widget\_dobav)  
zurnal.setGeometry(220, 178, 250, 240)  
zurnal.setStyleSheet('border-color: white; border-style: solid; border-width: 1px; font: 9pt "MS Shell Dlg 2";')  
zurnal.setAlignment(Qt.AlignTop)  
  
combo = QtWidgets.QComboBox(widget\_dobav)  
combo.setGeometry(220, 178, 250, 25)  
combo.hide()  
  
table = QtWidgets.QTableWidget(widget\_dobav)  
table.setStyleSheet("color: rgb(0, 0, 0);")  
rowPosition = table.rowCount()  
table.setColumnCount(1)  
table.setRowCount(4)  
table.setHorizontalHeaderLabels(['Диапазон, МГц'])  
table.setColumnWidth(0, 199)  
table.verticalHeader().hide()  
table.setGeometry(10, 178, 200, 240)  
widget\_dobav.hide()  
  
def DELETE\_TABLE():  
 name = lineName.text()  
 table\_1 = cur.execute("SELECT name FROM sqlite\_master WHERE type='table';").fetchall()  
 if (name,) in table\_1:  
 cur.execute('DROP TABLE IF EXISTS ' + str('"' + name + '"'))  
 conn.commit()  
 if ust == 0:  
 cur.execute('DELETE FROM reb WHERE name =' + str('"' + name + '"'))  
 conn.commit()  
 OBNOVA\_DATA()  
 index = form.comboBox.findText(name) # find the index of text  
 form.comboBox.removeItem(index)  
 msg = QMessageBox()  
 msg.setWindowTitle("Удаление")  
 msg.setText("Комплекс РЭБ удален")  
 msg.setIcon(QMessageBox.Information)  
 msg.exec\_()  
 else:  
 cur.execute('DELETE FROM antena WHERE name =' + str('"' + name + '"'))  
 conn.commit()  
 OBNOVA\_DATA()  
 index = form.comboBox.findText(name) # find the index of text  
 form.comboBox.removeItem(index)  
 msg = QMessageBox()  
 msg.setWindowTitle("Удаление")  
 msg.setText("Антенна удалена")  
 msg.setIcon(QMessageBox.Information)  
 msg.exec\_()  
 else:  
 msg = QMessageBox()  
 msg.setWindowTitle("Ошибка данных")  
 msg.setText("Такого устройства не существует")  
 msg.setIcon(QMessageBox.Warning)  
 msg.exec\_()  
  
pushDel.clicked.connect(DELETE\_TABLE)  
def OTOBRAZ():  
 table.clear()  
 table.setRowCount(0)  
 table.setColumnCount(1)  
 table.setHorizontalHeaderLabels(['Диапазон, МГц'])  
 diap = int(lineDiap.text())  
 rowPos = table.rowCount()  
 for i in range(diap):  
 table.insertRow(rowPos)  
  
pushOtobraz.clicked.connect(OTOBRAZ)  
def DOBAV\_REB\_OKNO():  
 global ust  
 ust = 0  
 label\_dobav.setText("Добавление комплекса РЭБ")  
 lineDiap.setText("")  
 lineName.setText("")  
 zurnal.setText("")  
 table.clear()  
 table.setColumnCount(1)  
 table.setRowCount(4)  
 table.setHorizontalHeaderLabels(['Диапазон, МГц'])  
 widget\_dobav.show()  
  
form.pushButton.clicked.connect(DOBAV\_REB\_OKNO)  
  
def DOBAV\_ANT\_OKNO():  
 global ust  
 ust = 1  
 label\_dobav.setText("Добавление антенны")  
 lineDiap.setText("")  
 lineName.setText("")  
 zurnal.setText("")  
 table.clear()  
 table.setColumnCount(1)  
 table.setRowCount(4)  
 table.setHorizontalHeaderLabels(['Диапазон, МГц'])  
 widget\_dobav.show()  
  
form.pushButton\_2.clicked.connect(DOBAV\_ANT\_OKNO)  
def DOBAV\_UST():  
 global ust  
 name = '"' + lineName.text() + '"'  
 table\_1 = cur.execute("SELECT name FROM sqlite\_master WHERE type='table';").fetchall()  
 reb = cur.execute("SELECT name FROM reb").fetchall()  
 ant = cur.execute("SELECT name FROM antena").fetchall()  
 if ust == 0:  
 if (str(name),) in table\_1 and (str(name),) in reb:  
 for i in range(int(lineDiap.text())):  
 item = str(table.item(int(i), 0).text())  
 cur.execute('INSERT INTO ' + name + ' (diapozon) VALUES (?)', (str(item),))  
 conn.commit()  
 zurnal.setText(zurnal.text() + "В " + str(name) + " добавлен " + str(item) + "\n")  
 else:  
 st = 'CREATE TABLE IF NOT EXISTS ' + name + ' (diapozon TEXT)'  
 cur.execute('CREATE TABLE IF NOT EXISTS ' + name + ' (diapozon TEXT)')  
 conn.commit()  
 for i in range(int(lineDiap.text())):  
 item = str(table.item(int(i), 0).text())  
 cur.execute('INSERT INTO ' + name + ' (diapozon) VALUES (?)', (str(item),))  
 conn.commit()  
 zurnal.setText(zurnal.text() + "В " + str(name) + " добавлен " + str(item) + "\n")  
 cur.execute('INSERT INTO reb (name) VALUES (?)', (str(name),))  
 conn.commit()  
 OBNOVA\_DATA()  
 else:  
 if (str(name),) in table\_1 and (str(name),) in ant:  
 for i in range(int(lineDiap.text())):  
 item = str(table.item(int(i), 0).text())  
 cur.execute('INSERT INTO ' + name + ' (diapozon) VALUES (?)', (str(item),))  
 conn.commit()  
 zurnal.setText(zurnal.text() + "В " + str(name) + " добавлен " + str(item) + "\n")  
 else:  
 cur.execute('CREATE TABLE IF NOT EXISTS ' + str(name) + ' (diapozon TEXT)')  
 conn.commit()  
 for i in range(int(lineDiap.text())):  
 item = str(table.item(int(i), 0).text())  
 print('INSERT INTO ' + name + ' (diapozon) VALUES (?)')  
 cur.execute('INSERT INTO ' + name + ' (diapozon) VALUES (?)', (str(item),))  
 conn.commit()  
 zurnal.setText(zurnal.text() + "В аненну" + str(name) + " добавлен " + str(item) + "\n")  
 cur.execute('INSERT INTO antena (name) VALUES (?)', (str(name),))  
 conn.commit()  
 OBNOVA\_DATA()  
 lineDiap.setText("")  
  
pushDobav.clicked.connect(DOBAV\_UST)  
#Время  
start\_time = 0  
stop\_time = 0  
  
def START\_ZURNAL():  
 global start\_time, stop\_time  
 start\_time = str(datetime.datetime.now().date()) + ' ' + str(datetime.datetime.now().time())  
 start\_time = start\_time[:-7]  
 form.label\_7.setText("Начало-" + str(start\_time[10:]))  
  
form.pushButton\_3.clicked.connect(START\_ZURNAL)  
  
def STOP\_ZURNAL():  
 global start\_time, stop\_time  
 stop\_time = str(datetime.datetime.now().date()) + ' ' + str(datetime.datetime.now().time())  
 stop\_time = stop\_time[:-7]  
 form.label\_7.setText("Конец-" + str(stop\_time[10:]))  
 cur.execute('INSERT INTO note (login, star, stopp) VALUES (?,?,?)', (str(form.login.text()), str(start\_time), str(stop\_time)))  
 conn.commit()  
  
form.pushButton\_4.clicked.connect(STOP\_ZURNAL)  
  
label = QLabel()  
label.resize(800, 700)  
label.setStyleSheet('background-color: rgb(70, 70, 70); color: rgb(255, 255, 255); border-color: black; border-style: solid; border-width: 5px; font: 12pt "MS Shell Dlg 2";')  
label.setAlignment(Qt.AlignTop | Qt.AlignHCenter)  
label.setWindowTitle("Журнал использования")  
  
def ZURNAL():  
 label.setText("Журнал использования\n\n")  
 row = cur.execute('SELECT \* FROM note')  
 count = 1  
 for i in row:  
 label.setText(label.text() + str(count) + ". Пользователь: " + str(i[1]) + "; Начало: " + str(i[2]) + "; Конец: " + str(i[3]) + "\n")  
 count += 1  
 label.show()  
  
form.pushButton\_9.clicked.connect(ZURNAL)  
  
def OCHITKA\_ZURNALA():  
 cur.execute('DELETE FROM note')  
 conn.commit()  
 label.setText("Журнал использования\n\n")  
  
form.pushButton\_8.clicked.connect(OCHITKA\_ZURNALA)  
  
window.show()  
app.exec()