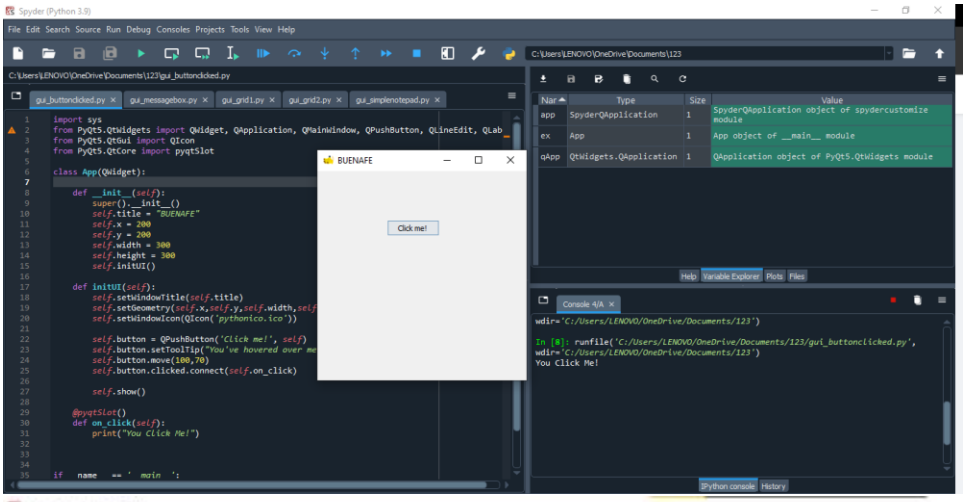
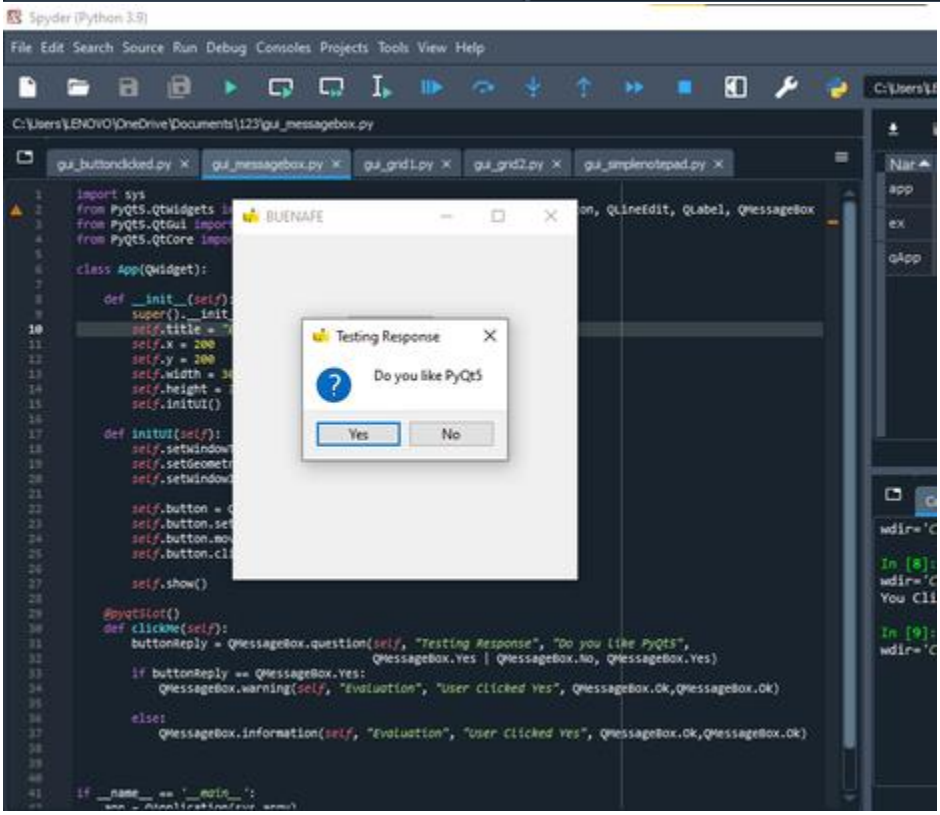
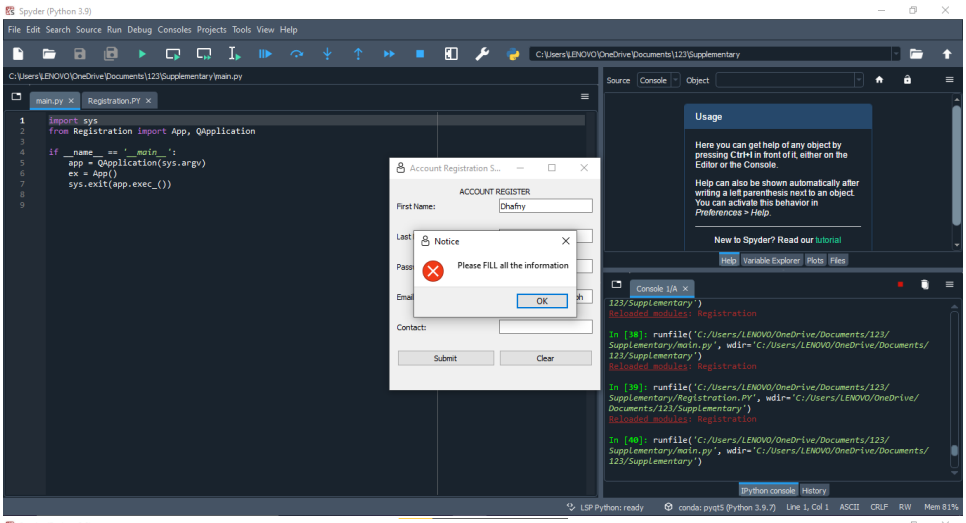
Hands-on Activity 6.2 Introduction to Event Handling in GUI Development	
Buenafe, Dhafny Francisco, Lauper Xavier Efa, Christian Ed	4/20/2022
BSCPE12S1	Engr. Royce B. Chua

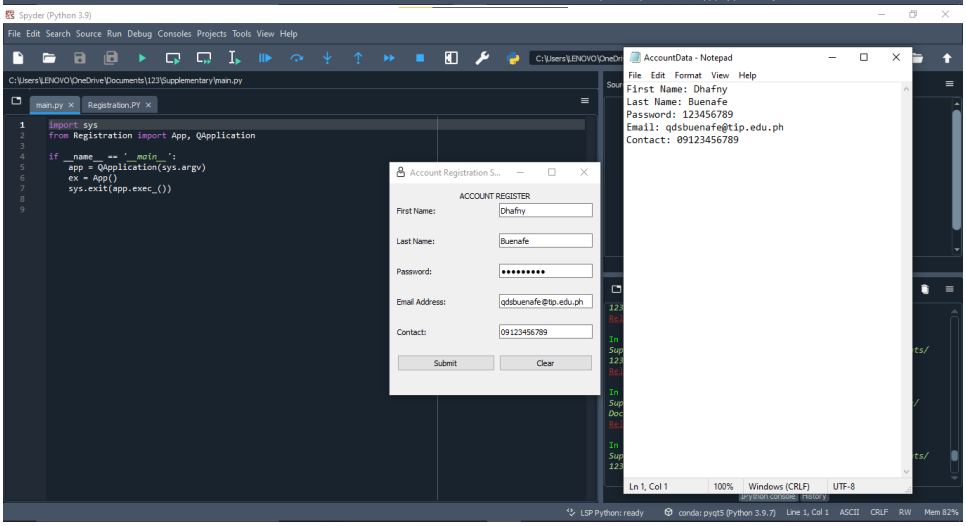
Procedures:





Supplementary Task:





main.py × Registration.PY ×

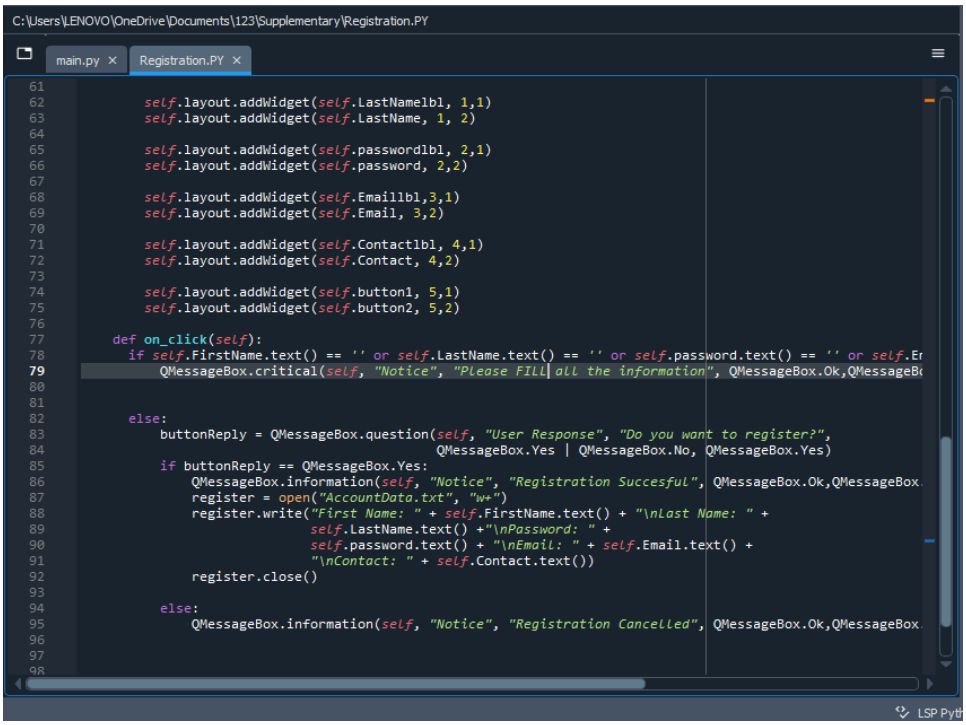
```
from PyQt5.QtWidgets import QWidget, QApplication, QPushButton, \
QLineEdit, QLabel, QGridLayout, QMessageBox
from PyQt5.QtGui import QIcon

class App(QWidget):

    def __init__(self):
        super().__init__()
        self.title = "Account Registration System"
        self.x = 550
        self.y = 250
        self.width = 300
        self.height = 300
        self.initUI()

    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(self.x,self.y,self.width,self.height)
        self.setWindowIcon(QIcon('pythonico.ico'))

        self.createGridLayout()
        self.setLayout(self.layout)
        self.show()

    def createGridLayout(self):
        self.layout = QGridLayout()

        self.layout.setColumnStretch(1,2)

        self.textboxlbl = QLabel("ACCOUNT REGISTER",self)
        self.textboxlbl.move(100,2)


        self.FirstNamelbl = QLabel("First Name: ", self)
        self.FirstName = QLineEdit(self)

        self.LastNamelbl = QLabel("Last Name: ", self)
```

C:\Users\LENOVO\OneDrive\Documents\123\Supplementary\Registration.PY

main.py × Registration.PY ×

```
        self.LastNamelbl = QLabel("Last Name: ", self)
        self.LastName = QLineEdit(self)

        self.passwordlbl = QLabel("Password: ", self)
        self.password = QLineEdit(self)
        self.password.setEchoMode(QLineEdit.Password)

        self.Emaillbl = QLabel("Email Address: ", self)
        self.Email = QLineEdit(self)


        self.Contactlbl = QLabel("Contact: " ,self)
        self.Contact = QLineEdit(self)



        self.button1 = QPushButton('Submit', self)
        self.button1.setToolTip("")
        self.button1.clicked.connect(self.on_click)
        self.button2 = QPushButton('Clear', self)
        self.button2.setToolTip("")

        self.layout.addWidget(self.FirstNamelbl, 0,1)
        self.layout.addWidget(self.FirstName, 0,2)

        self.layout.addWidget(self.LastNamelbl, 1,1)
        self.layout.addWidget(self.LastName, 1, 2)

        self.layout.addWidget(self.passwordlbl, 2,1)
        self.layout.addWidget(self.password, 2,2)

        self.layout.addWidget(self.Emaillbl,3,1)
        self.layout.addWidget(self.Email, 3,2)

        self.layout.addWidget(self.Contactlbl, 4,1)
        self.layout.addWidget(self.Contact, 4,2)

        self.layout.addWidget(self.button1, 5,1)
```

LSP Pyt



Guide Questions:

1. button_is_checked on self. This allows you to work with the values like any other Python variable and without accessing the original widget.

.windowTitleChanged signal. This signal is emitted when the dock widget window's title has changed.

Clicked signal is emitted when a mouse button is clicked

2. For me Event handling is divided into signal and slot is to make it easier to understand because the signal connects to the slot by calling the method where the slot is.

3. Message box can be used to guide the users GUI application so that users won't be having problems in using the application. Message boxes is very useful in programming specially in creating GUI applications since most of the times user is not familiar in using a certain application. Everyone especially the users benefit in using message boxes since it displays messages that will be needed in using the application which usually contains instructions about the applications or just a message.

4. Error handling refers to how a software program responds to and recovers from errors. It anticipates, detects, and resolves application, programming, and communication issues. Error handling keeps the program flow regularly. While many apps confront several design issues

when dealing with errors, Error handling gently handles hardware and software failures and helps restart operations when stopped. Error handling in software is done either by the programmer writing the code or by using software tools. When mistakes cannot be categorized, they are handled by returning specific error codes.

5. Error handling is important because it helps end users utilize your code properly. It also makes code easier to maintain.

Conclusion:

we conclude that after answering and solving the problems on this lab activity we are able to identify the main components in a gui application and create a simple gui application using pyqt5 widgets. we are having a difficulties with the supplementary task but we solved it by a method to create a text file. by coding we encounter problems such as error in indentions,spelling, and missing file. we are checking each of our codes and solving the problem to execute the program and run it properly.

"I affirm that I have not given or received any unauthorized help on this assignment, and that this work is my own."