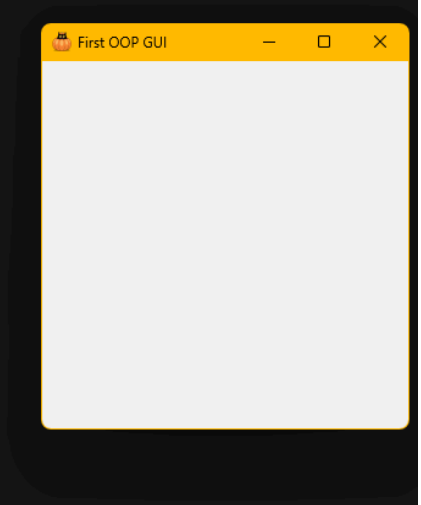
Activity #4 - Introduction to GUI Development using Pycharm	
Magistrado, Aira Pauleen M.	10/14/24
CPE009/CPE21S4	Maria Rizette Sayo

Procedure:

1) Adding Icon:

gui_buttons.py gui_labels.py gui_text.py lab4.py

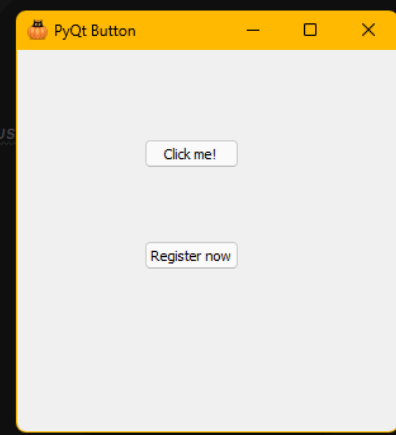
```python
import sys
from PyQt5.QtWidgets import QMainWindow, QApplication
from PyQt5.QtGui import QIcon

1 usage
class App(QMainWindow):

    def __init__(self):
        super().__init__() #Initialize the main window like the previous one
        # window = QMainwIndow()
        self.title = "First OOP GUI"
        self.initUI()

    1 usage
    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(200,200,300,300)
        self.setWindowIcon(QIcon('pythonico.ico')) #sets an icon
        self.show()

if __name__ == '__main__':
    app = QApplication(sys.argv)
    Main = App()
    sys.exit(app.exec_())
```



2) Creating Buttons:

gui_buttons.py × gui_labels.py gui_text.py lab4.py

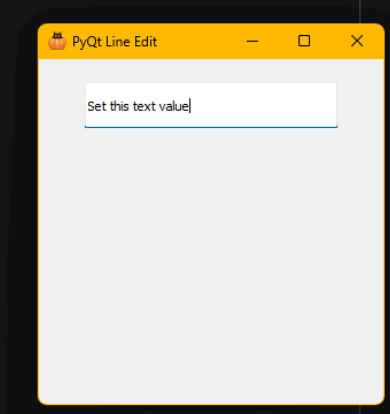
```python
import sys
from PyQt5.QtWidgets import QWidget,QApplication,QMainWindow, QPushButton
from PyQt5.QtGui import QIcon

1 usage
class App(QWidget):

    def __init__(self):
        super().__init__() #Initialize the main window like the previous
        # window = QMainwIndow()
        self.title = "PyQt Button"
        self.x=200 #or left
        self.y=200 #or top
        self.width=300
        self.height=300
        self.initUI()

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

        #In GUI Oython, these buttons, textboxes, labels are called Widgets
        #first button
        self.button = QPushButton('Click me!', self)
        self.button.setToolTip("You've hovered over me!")
        self.button.move(100,70) #button.move(x,y)

        #second button
        self.button2 = QPushButton('Register now', self)
        self.button2.setToolTip("Click the button to register!")
        self.button2.move(100, 150)
        self.show()

if __name__ == '__main__':
    app = QApplication(sys.argv)
    Main = App()
    sys.exit(app.exec_())
```



3) Creating Text Fields

gui_buttons.py gui_labels.py gui_text.py lab4.py

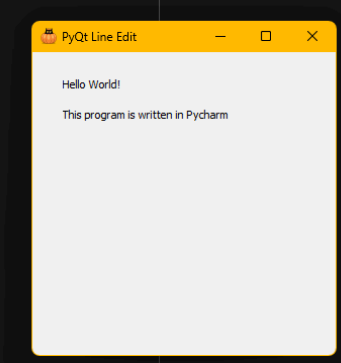
```python
import sys
from PyQt5.QtWidgets import QWidget, QApplication, QLineEdit
from PyQt5.QtGui import QIcon

1 usage
class App(QWidget):

    def __init__(self):
        super().__init__() #initializes the main window like in previous one
        #window = QMainWindow()
        self.title = "PyQt Line Edit"
        self.x=200 #or left
        self.y=200 #or top
        self.width=300
        self.height=300
        self.initUI()

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

    #Create Textbox
        self.textbox = QLineEdit(self)
        self.textbox.move(40,20)
        self.textbox.resize(220,40)
        self.textbox.setText("Set this text value")

        self.show()

if __name__ == '__main__':
    app = QApplication(sys.argv)
    Main = App()
    sys.exit(app.exec_())
```



4) Creating Labels

gui_buttons.py gui_labels.py gui_text.py lab4.py

```python
import sys
from PyQt5.QtWidgets import QWidget, QApplication, QMainWindow, QPushButton, QLineEdit, QLabel
from PyQt5.QtGui import QIcon

1 usage
class App(QWidget):
    def __init__(self):
        super().__init__()  # initializes the main window like in previous one
    # window = QMainWindow()
        self.title = "PyQt Line Edit"
        self.x = 200  # or left
        self.y = 200  # or top
        self.width = 300
        self.height = 300
        self.initUI()

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

        #1st text
        self.textboxbl = QLabel("Hello World!", self)
        self.textboxbl.move(30, 25)

        #2nd text
        self.textboxbl2 = QLabel("This program is written in Pycharm", self)
        self.textboxbl2.move(30, 55)

        self.show()

if __name__ == '__main__':
    app = QApplication(sys.argv)
    Main = App()
    sys.exit(app.exec_())
```



Supplementary Activity:

registration.py

```python
from PyQt5.QtWidgets import QMainWindow, QApplication, QLabel, QLineEdit, QPushButton

from PyQt5.QtGui import QIcon

from PyQt5.QtCore import Qt


class RegistrationApp(QMainWindow):


    def __init__(self):

        super().__init__()

        self.title = "Account Registration Form"

        self.width = 400
```

```
        self.height = 400
        self.initUI()

    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(100, 100, self.width, self.height)
        self.setWindowIcon(QIcon('pythonico.ico'))
        self.center()

        self.program_title = QLabel("Account Registration", self)
        self.program_title.setAlignment(Qt.AlignCenter)
        self.program_title.setGeometry(0, 10, self.width, 30)

        self.label_first_name = QLabel("First Name:", self)
        self.label_first_name.move(30, 50)
        self.textbox_first_name = QLineEdit(self)
        self.textbox_first_name.move(150, 50)
        self.textbox_first_name.resize(200, 30)

        self.label_last_name = QLabel("Last Name:", self)
        self.label_last_name.move(30, 90)
        self.textbox_last_name = QLineEdit(self)
        self.textbox_last_name.move(150, 90)
        self.textbox_last_name.resize(200, 30)

        self.label_username = QLabel("Username:", self)
        self.label_username.move(30, 130)
        self.textbox_username = QLineEdit(self)
        self.textbox_username.move(150, 130)
        self.textbox_username.resize(200, 30)

        self.label_password = QLabel("Password:", self)
```

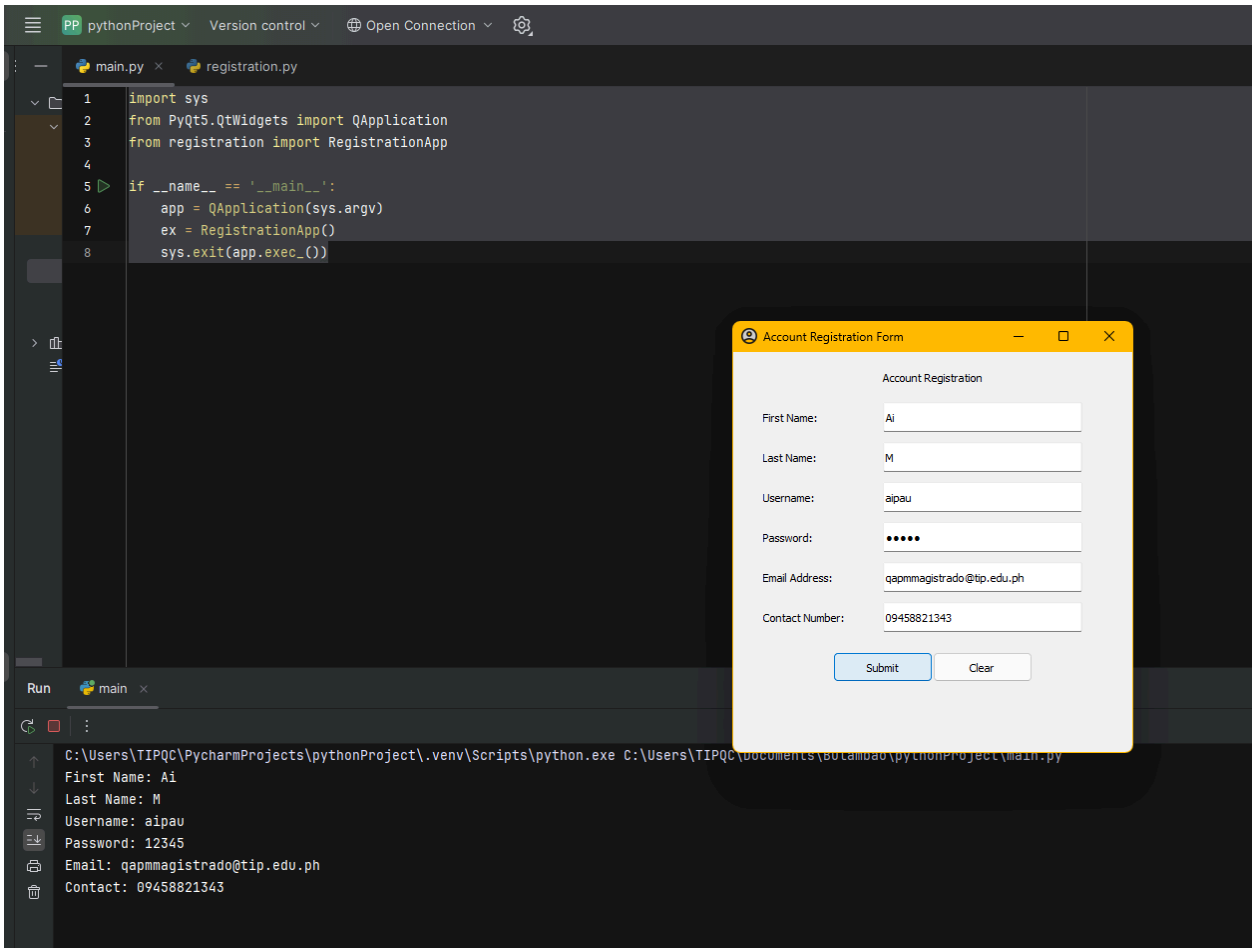
```
        self.label_password.move(30, 170)
        self.textbox_password = QLineEdit(self)
        self.textbox_password.move(150, 170)
        self.textbox_password.resize(200, 30)
        self.textbox_password.setEchoMode(QLineEdit.Password)

        self.label_email = QLabel("Email Address:", self)
        self.label_email.move(30, 210)
        self.textbox_email = QLineEdit(self)
        self.textbox_email.move(150, 210)
        self.textbox_email.resize(200, 30)

        self.label_contact = QLabel("Contact Number:", self)
        self.label_contact.move(30, 250)
        self.textbox_contact = QLineEdit(self)
        self.textbox_contact.move(150, 250)
        self.textbox_contact.resize(200, 30)

        self.submit_button = QPushButton('Submit', self)
        self.submit_button.setToolTip("Click to submit")
        self.submit_button.move(100, 300)
        self.submit_button.clicked.connect(self.submit)

        self.clear_button = QPushButton('Clear', self)
        self.clear_button.setToolTip("Click to clear")
        self.clear_button.move(200, 300)
        self.clear_button.clicked.connect(self.clear)

        self.show()

    def center(self):
        qr = self.frameGeometry()
```

```
        cp = QApplication.desktop().availableGeometry().center()
        qr.moveCenter(cp)
        self.move(qr.topLeft())

    def submit(self):
        first_name = self.textbox_first_name.text()
        last_name = self.textbox_last_name.text()
        username = self.textbox_username.text()
        password = self.textbox_password.text()
        email = self.textbox_email.text()
        contact = self.textbox_contact.text()

        print(f"First Name: {first_name}")
        print(f"Last Name: {last_name}")
        print(f"Username: {username}")
        print(f"Password: {password}")
        print(f"Email: {email}")
        print(f"Contact: {contact}")

    def clear(self):
            self.textbox_first_name.clear()
            self.textbox_last_name.clear()
            self.textbox_username.clear()
            self.textbox_password.clear()
            self.textbox_email.clear()
            self.textbox_contact.clear()
```

main.py

```
import sys
from PyQt5.QtWidgets import QApplication
```

```python
from registration import RegistrationApp


if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = RegistrationApp()
    sys.exit(app.exec_())
```



Questions

1. What are the common GUI Applications that general end-users such as home users, students, and office employees use? (give at least 3 and describe each)

The common GUI applications are Google Chrome, a user-friendly web browser for research and watching instructional or entertaining videos, watch educational or entertainment videos, Zoom, which gained popularity during the quarantine period when classes were conducted virtually and online meetings were used, and Canva, which allows users to create designs without the need for graphic design expertise because it offers a wide selection of templates for both personal and professional use.

2. Based from your answer in question 1, why do you think home users, students, and office employees use those GUI programs?

These types of graphical user interface (GUI) programs are used by a variety of users because they are simple to use, convenient for all users, and can be accessed from a variety of devices. They are designed with accessibility and usability in mind.

3. How does Pycharm help developers in making GUI applications, what would be the difference if developers made GUI programs without GUI Frameworks such as Pycharm or Tkinter?

PyCharm makes it easier for developers to create programs that have graphical user interfaces (GUIs). It offers features like code suggestions, debugging tools, and error highlighting. These features reduce errors and speed up the debugging process. Without the availability of tools such as PyCharm, developers would have to perform time-consuming and tedious manual tasks like creating and managing windows. These tasks are made easier by programs like PyCharm, which offer pre-built features and components.

4. What are the different platforms a GUI program may be created and deployed on? (Three is required then state why might a program be created on that specific platform).

The different platform that GUI programs may be created or employed are Windows because it can reach a wide range of users and companies use it a lot. Building and deploying apps is made simpler by the platform's exceptional development tools. MacOS renowned for its superior user experience and high performance. Especially for users in creative and professional fields, developers choose macOS to ensure their applications meet high standards and offer a polished, seamless experience. and Android which allows your application to reach a global audience because of its large user base and support for a wide variety of devices.

5. What is the purpose of app = QApplication(sys.argv), ex = App(), and sys.exit(app.exec_())?

The function of app = QApplication(sys.argv) is to start your app in the code. It is the one in charge of setting things up so your program can run as GUI. ex = App() is in charge of the main window for your app. It's the foundation where everything else in your app will go. sys.exit(app.exec_()) is in charge of the main loop of the app. It runs your app and then safely shuts it down when you're done.

Conclusion:

In conclusion, In this lab activity, I was introduced to and learned the PyCharm framework for GUI development. Like how to add icons, buttons, text fields, and labels. In the supplementary, I was tasked to make a simple Account Registration System from scratch. My understanding of design and usability was improved by adding buttons, centering the GUI, and aligning text fields and labels to the basic account registration system. My knowledge of PyCharm and Python's capabilities for building interactive programs has grown, and the different procedures have given me the skills and capability to create GUI applications.