Laboratory Activity No. 4 - Introduction to GUI Development using Pycharm	
Justin Bjorn L. Sanchez	**10/16/2024**
CPE21S1 / CPE009	**Mrs. Maria Rizette Sayo**

Main.py

```
import sys
from PyQt5.QtWidgets import QWidget, QLineEdit,
QPushButton, QApplication, QLabel
from PyQt5.QtGui import QIcon
from registration import App

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

Registration.py

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


class App(QWidget):

   def __init__(self):
       super().__init__()
       self.title = "PyQt Registration"
       self.x = 200
       self.y = 200
       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.textboxlbl = QLabel("Registration", self)
       self.textboxlbl.move(120, 20)

       self.textboxlbl2 = QLabel("First name:", self)
       self.textboxlbl2.move(20, 70)
       self.textboxentry = QLineEdit(self)
       self.textboxentry.move(80, 70)

       self.textboxlbl3 = QLabel("Last name:", self)
       self.textboxlbl3.move(20, 100)
       self.textboxentry2 = QLineEdit(self)
```

| | ```
        self.textboxentry2.move(80, 100)

        self.textboxlbl4 = QLabel("Username:", self)
        self.textboxlbl4.move(20, 130)
        self.textboxentry3 = QLineEdit(self)
        self.textboxentry3.move(80, 130)

        self.textboxlbl5 = QLabel("Password:", self)
        self.textboxlbl5.move(20, 160)
        self.textboxentry4 = QLineEdit(self)
        self.textboxentry4.move(80, 160)
        self.textboxentry4.setEchoMode(QLineEdit.Password)

        self.textboxlbl6 = QLabel("Email:", self)
        self.textboxlbl6.move(20, 190)
        self.textboxentry5 = QLineEdit(self)
        self.textboxentry5.move(80, 190)

        self.button = QPushButton('Submit', self)
        self.button.clicked.connect(self.submit)
        self.button.move(50, 230)

        self.result_label = QLabel("", self)
        self.result_label.move(80, 280)

        self.button2 = QPushButton('Clear', self)
        self.button2.move(150, 230)
        self.button2.clicked.connect(self.clear)

    def submit(self):
        first_name = self.textboxentry.text()
        last_name = self.textboxentry2.text()
        username = self.textboxentry3.text()
        password = self.textboxentry4.text()
        email = self.textboxentry5.text()
        self.result_label.setText("Registration
successful!")
    def clear(self):
        self.textboxentry.clear()
        self.textboxentry2.clear()
        self.textboxentry3.clear()
        self.textboxentry4.clear()
        self.textboxentry5.clear()


if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = App()
    sys.exit(app.exec_())
``` |
| GUI Labels | ```
import sys
``` |

	`` from PyQt5.QtWidgets import QWidget, QLineEdit, QMainWindow, QApplication, QLabel from PyQt5.QtGui import QIcon class App(QWidget): def __init__(self): super().__init__() self.title = "PyQt Line Edit" self.x = 200 self.y = 200 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.textboxlbl = QLabel("Hello world!", self) self.textboxlbl.move(100, 150) self.textboxlbl = QLabel("This program is written in Pycharm", self) self.textboxlbl.move(100, 170) self.show() if __name__ == '__main__': app = QApplication(sys.argv) ex = App() sys.exit(app.exec_()) ```
GUI Text	``` import sys from PyQt5.QtWidgets import QWidget, QLineEdit, QMainWindow, QApplication from PyQt5.QtGui import QIcon class App(QWidget): def __init__(self): super().__init__() self.title = "PyQt Line Edit" self.x = 200 self.y = 200 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.textbox = QLineEdit(self) self.textbox.move(20, 20) self.textbox.resize(280, 40) self.textbox.setText("Set this text value") self.show() if __name__ == '__main__': app = QApplication(sys.argv) ex = App() sys.exit(app.exec_()) ```
GUI Buttons	``` import sys from PyQt5.QtWidgets import QWidget, QPushButton, QMainWindow, QApplication from PyQt5.QtGui import QIcon class App(QWidget): def __init__(self): super().__init__() self.title = "PyQt Button" self.x = 200 self.y = 200 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.button = QPushButton('Click me!', self) self.button.setToolTip("You've hovered over me") self.button.move(100, 70) self.button2 = QPushButton('Register', self) self.button2.setToolTip('This button does nothing... yet') self.button2.move(100, 95) self.show() if __name__ == '__main__': app = QApplication(sys.argv) ex = App() sys.exit(app.exec_()) ```

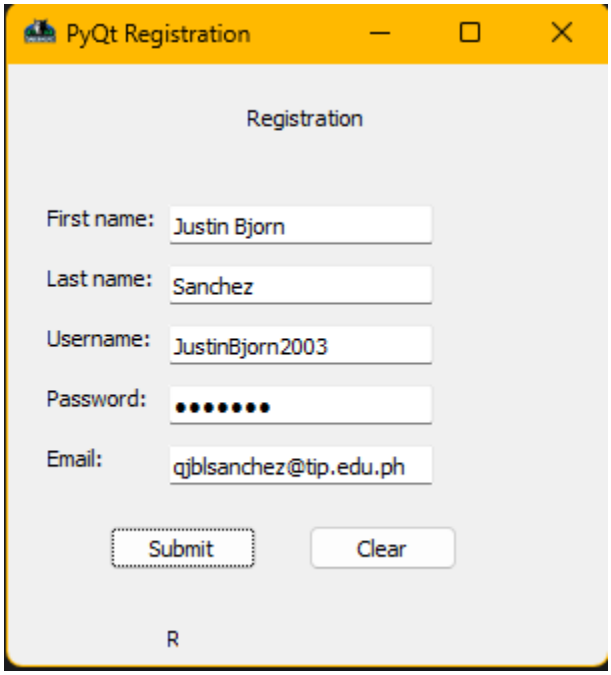
Class app	
INTERFACE OUTPUT	

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


class App(QMainWindow):
    def __init__(self):
        super().__init__()
        self.title = "First OOP GUI"
        self.initUI()

    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(200, 200, 300, 300)
        self.setWindowIcon(QIcon('pythonico.ico'))
        self.show()


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



QUESTIONS:

1. Common GUI Applications Used by Home Users, Students, and Office Employees:

a. Microsoft Office Suite (Word, Excel, PowerPoint):
These applications are essential for creating documents, spreadsheets, and presentations. They are commonly used in academic, personal, and professional settings for tasks like writing reports, managing data, and preparing slideshows.

b. Web Browsers (Google Chrome, Firefox, Microsoft Edge):
Web browsers are used for accessing the internet. They are vital tools for researching, browsing websites, using web applications, and staying connected through social media or emails.

c. Video Conferencing Tools (Zoom, Microsoft Teams, Google Meet):
These applications are used to communicate via video calls. Students use them for online classes, office workers for remote meetings, and home users to connect with family and friends.

2. Why Do Home Users, Students, and Office Employees Use These GUI Programs?

a. Microsoft Office Suite:
It's versatile and widely recognized. Students use it for school projects, office workers for creating professional documents and presentations, and home users for personal projects or small business needs. Its ease of use and essential features make it a must-have.

b. Web Browsers:
The internet is a major source of information, entertainment, and communication. Students need browsers for research and online classes, office workers use them for accessing cloud-based tools, and home users rely on them for social media and online shopping.

c. Video Conferencing Tools:
These tools are essential in today's world where remote communication is common. They provide a way to connect, share screens, and collaborate from different locations, which is crucial for both work and education.

3. How Does PyCharm Help Developers in Making GUI Applications? What Would Be the Difference Without GUI Frameworks?

PyCharm:
PyCharm is an IDE that makes coding easier. It has features like auto-completion, debugging tools, and easy integration with GUI frameworks like Tkinter or PyQt. This makes building graphical applications more efficient since the IDE supports you in writing, testing, and managing your code.

Without GUI Frameworks:
Without a framework, developers would need to handle the low-level details of drawing the interface (like making buttons and windows) and managing user interactions. This would take a lot of time and effort. Frameworks simplify this by providing pre-made components like buttons and text boxes that you

can easily include in your program.

4. Different Platforms for Creating and Deploying GUI Programs:

a. Windows:
Windows is popular for home users and businesses, so creating GUI applications for this platform ensures a large user base. Many users prefer Windows for its compatibility with a variety of software.

b. macOS:
macOS is widely used by professionals, especially in the fields of design and media. GUI programs created for macOS benefit from the system's clean interface and integration with creative software.

c. Linux:
Linux is used by developers and researchers because it's open-source and highly customizable. Creating applications for Linux can attract tech-savvy users or be used in specialized environments like servers and research labs.

5. Purpose of app = QApplication(sys.argv), ex = App(), and sys.exit(app.exec_()):

a.app = QApplication(sys.argv):
This line sets up the application by creating an instance of QApplication. It initializes the GUI and processes command-line arguments if needed. Without this, the program wouldn't be able to handle GUI events.

b. ex = App():
This creates an instance of the main application window (App class). This is where the GUI elements are defined, like buttons and text boxes. It sets up the interface for users to interact with.

c. sys.exit(app.exec_()):
This starts the event loop, which is responsible for handling user inputs like mouse clicks or key presses. Once the user closes the application, it exits the event loop and ensures the program terminates properly by calling sys.exit().

Conclusion:

Through this laboratory activity, I gained hands-on experience with creating basic GUI applications using PyCharm. I learned how to add various widgets like buttons, text fields, and labels, as well as how to organize them using absolute positioning. This helped me understand the importance of frameworks like PyQt for simplifying GUI development, as well as the role of PyCharm in making the coding process more efficient. Overall, this activity enhanced my understanding of GUI components and how they are structured within a Python application.