



Homework 2

Software Engineering Principle

Software Engineering Program,

Department of Computer Engineering,

School of Engineering, KMITL

67011287 Ramida Laphasphokin

Features

Gantt Chart

Used for planning project plan and tasks duration or deadline.

Class Diagram

Used for designing classes in the projects.

Interaction Diagram

Used for designing how classes interact each others in the projects.

Markdown Renderer for the task assignment page

How it works:

- As mentioned earlier, we can use markdown to express the task, ∴ we need a markdown renderer.

Implementation Approach:

- Use QEngineWebView Module in PyQt.

VS-Code Extension (OPTIONAL)

TODO extension in VS-Code with better description for the task and with team member(s) assigned to that task.

How it works:

- If you have comment with TODO in the front, you can add description of the task in a different entry and also in a markdown file.
- If you want to add a person in charge for that task (OPTIONAL), you can use @TEMP, where TEMP can be either role or team member names.
- After saved, you can access the TODO description as you hover and click to inspect task in the comment.

Implementation Approach:

- Scan through the file looking for comment with TODO in the front then keep the entry into the DB.
- We can edit the TODO description inside a external markdown file.

Page included in this homework

- **Design Page:** Page to brainstorm the program architecture and to adjust tasks duration or deadline.
- **User Settings Page:** Page to edit current user settings.

Code:

DesignPage.py

```
import sys
from PySide6.QtWidgets import *
from PySide6.QtCore import *
from PySide6.QtGui import *

class Design(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Design Board")
        self.setMinimumSize(1280, 720)
        self.setStyleSheet("color: black; background-color:
                           whitesmoke;")

    mainLayout = QVBoxLayout(self)

    navbar = QWidget()
    navbar.setFixedHeight(100)
    navbar.setStyleSheet("""
        background-color: #2f8ed2;
        color: white;
    """)
    navLayout = QHBoxLayout(navbar)
    navLayout.setContentsMargins(10, 0, 10, 0)

    btnClass = QPushButton("Class")
    btnInteraction = QPushButton("Interaction")
    btnGantt = QPushButton("Gantt Chart")

    navLayout.addStretch()

    for b in (btnClass, btnInteraction, btnGantt):
        b.setFixedWidth(200)
        b.setFixedHeight(100)
        b.setStyleSheet("""
            QPushButton {
                font-size: 16px;
                background: #2f8ed2;
            }
        """)


```

```

                color: white;
            }
            QPushButton:hover {
                background: #fdd148;
                color: black;
            }
        """
    )
    navLayout.addWidget(b, alignment=Qt.AlignCenter)

navLayout.addStretch()

mainLayout.addWidget(navbar)

# ===== TOOL PANEL (LEFT SIDE OF BOARD) =====
self.toolsPanel = QWidget()
self.toolsPanel.setFixedWidth(150)
self.toolsPanel.setStyleSheet("""
    background-color: #d9ecf7;
    border-right: 1px solid #b0cddd;
""")
toolsLayout = QVBoxLayout(self.toolsPanel)
toolsLayout.setContentsMargins(10, 20, 10, 20)
toolsLayout.setSpacing(10)

toolsLayout.addWidget(QLabel("Tools"))
toolsLayout.addWidget(QPushButton("Tool 1"))
toolsLayout.addWidget(QPushButton("Tool 2"))
toolsLayout.addWidget(QPushButton("Tool 3"))
toolsLayout.addStretch()

self.boardStack = QStackedWidget()

# --- Class Board ---
classBoard = QWidget()
classLayout = QVBoxLayout(classBoard)
classTitle = QLabel("CLASS DIAGRAM BOARD")
classTitle.setAlignment(Qt.AlignCenter)
classTitle.setStyleSheet("font-size: 22px; font-weight:
    ↳ bold;")
classLayout.addWidget(classTitle)

```

```

classImg = QLabel("Class Diagram")
classImg.setMinimumHeight(450)
classImg.setAlignment(Qt.AlignCenter)
classImg.setStyleSheet("""
    background: #eeeeee;
    border: 2px dashed #999;
""")
classLayout.addWidget(classImg)
classLayout.addStretch()

# --- Interaction Board ---
interactionBoard = QWidget()
interactionLayout = QVBoxLayout(interactionBoard)
interactionTitle = QLabel("INTERACTION DIAGRAM BOARD")
interactionTitle.setAlignment(Qt.AlignCenter)
interactionTitle.setStyleSheet("font-size: 22px;
    ↳ font-weight: bold; ")
interactionLayout.addWidget(interactionTitle)

interactImg = QLabel("Interaction Diagram")
interactImg.setMinimumHeight(450)
interactImg.setAlignment(Qt.AlignCenter)
interactImg.setStyleSheet("""
    background: #eeeeee;
    border: 2px dashed #999;
""")
interactionLayout.addWidget(interactImg)
interactionLayout.addStretch()

# --- Gantt Chart Board ---
ganttBoard = QWidget()
ganttLayout = QVBoxLayout(ganttBoard)
ganttTitle = QLabel("GANNT CHART")
ganttTitle.setAlignment(Qt.AlignCenter)
ganttTitle.setStyleSheet("font-size: 22px; font-weight:
    ↳ bold; ")
ganttLayout.addWidget(ganttTitle)

ganttImg = QLabel("Gantt Chart")
ganttImg.setMinimumHeight(450)

```

```

ganttImg.setAlignment(Qt.AlignCenter)
ganttImg.setStyleSheet("""
    background: #eeeeee;
    border: 2px dashed #999;
""")
ganttLayout.addWidget(ganttImg)
ganttLayout.addStretch()

self.boardStack.addWidget(classBoard)
self.boardStack.addWidget(interactionBoard)
self.boardStack.addWidget(ganttBoard)

contentWrapper = QBoxLayout()
contentWrapper.addWidget(self.toolsPanel)
contentWrapper.addWidget(self.boardStack)

wrapperWidget = QWidget()
wrapperWidget.setLayout(contentWrapper)
mainLayout.addWidget(wrapperWidget)

btnClass.clicked.connect(lambda: self.switchBoard(0))
btnInteraction.clicked.connect(lambda: self.switchBoard(1))
btnGantt.clicked.connect(lambda: self.switchBoard(2))

self.show()

def switchBoard(self, index):
    self.boardStack.setCurrentIndex(index)

    if index == 2:
        self.toolsPanel.hide()
    else:
        self.toolsPanel.show()

if __name__ == '__main__':
    app = QApplication(sys.argv)

    w = Design()

    sys.exit(app.exec())

```

UserSettingsPage.py

```
import sys
from PySide6.QtWidgets import *
from PySide6.QtCore import *
from PySide6.QtGui import *

class Setting(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Setting")
        self.setMinimumSize(1280, 720)

        mainLayout = QBoxLayout()
        self.setLayout(mainLayout)
        self.setStyleSheet("color: black; background-color:
                           whitesmoke;")

        col1 = QVBoxLayout()
        col1.setContentsMargins(20, 20, 20, 20)
        col1.setSpacing(15)

        title = QLabel("Account:")
        title.setFont(QFont("Arial", 28, QFont.Bold))
        title.setContentsMargins(0, 0, 0, 20)
        col1.addWidget(title)

        leftWidget = QWidget()
        leftWidget.setLayout(col1)
        leftWidget.setStyleSheet("background-color: #fdd148;
                               border-radius: 10px;")

        mainLayout.addWidget(leftWidget, 1)

        profilePic = QLabel()
        profilePic.setFixedSize(200, 200)
        profilePic.setStyleSheet("background: gray; border-radius:
                               100px;")
        col1.addWidget(profilePic, alignment=Qt.AlignCenter)

        self.usernameLabel = QLabel("Username: Raph")
```

```

self.gmailLabel = QLabel("Email: raph@example.com")
self.idLabel = QLabel("Account ID: 12345")
self.phoneLabel = QLabel("Phone: +66 1234 5678")
self.statusLabel = QLabel("Status: Active")

info_labels = [
    self.usernameLabel,
    self.gmailLabel,
    self.phoneLabel,
    self.idLabel,
    self.statusLabel,
]
for label in info_labels:
    label.setFont(QFont("Arial", 16))
    label.setAlignment(Qt.AlignCenter)
    col1.addWidget(label)

col1.addStretch()

scrollArea = QScrollArea()
scrollArea.setWidgetResizable(True)

scrollArea.setStyleSheet("background-color: #f4f4f4;")

formContainer = QWidget()
formLayout = QVBoxLayout()
formLayout.setContentsMargins(20, 20, 20, 20)
formLayout.setSpacing(20)
formContainer.setLayout(formLayout)

scrollArea.setWidget(formContainer)
scrollArea.setStyleSheet("font-size: 16px")
mainLayout.addWidget(scrollArea, 2)

formTitle = QLabel("Edit Account Information")
formTitle.setStyleSheet("font-size: 24px; font-weight:
    ↪ bold;")
formLayout.addWidget(formTitle)

formLayout.addWidget(QLabel("Username:"))

```

```

self.usernameEdit = QLineEdit()
self.usernameEdit.setPlaceholderText("Enter new username")
formLayout.addWidget(self.usernameEdit)

formLayout.addWidget(QLabel("Email:"))
self.gmailEdit = QLineEdit()
self.gmailEdit.setPlaceholderText("Enter new email")
formLayout.addWidget(self.gmailEdit)

formLayout.addWidget(QLabel("Phone:"))
self.phoneEdit = QLineEdit()
self.phoneEdit.setPlaceholderText("Enter phone number")
formLayout.addWidget(self.phoneEdit)

formLayout.addWidget(QLabel("Account ID (Read-only):"))
self.idEdit = QLineEdit("12345")
self.idEdit.setReadOnly(True)
formLayout.addWidget(self.idEdit)

cpTitle = QLabel("Change Password")
cpTitle.setStyleSheet("font-size: 24px; font-weight:
    ↳ bold;")
cpTitle.setContentsMargins(0, 20, 0, 10)
formLayout.addWidget(cpTitle)

self.oldPassword = QLineEdit()
self.oldPassword.setEchoMode(QLineEdit.Password)
self.oldPassword.setPlaceholderText("Enter old password")

self.newPassword = QLineEdit()
self.newPassword.setEchoMode(QLineEdit.Password)
self.newPassword.setPlaceholderText("Enter new password")

self.confirmPassword = QLineEdit()
self.confirmPassword.setEchoMode(QLineEdit.Password)
self.confirmPassword.setPlaceholderText("Confirm new
    ↳ password")

formLayout.addWidget(QLabel("Old Password:"))
formLayout.addWidget(self.oldPassword)

```

```

formLayout.addWidget(QLabel("New Password:"))
formLayout.addWidget(self.newPassword)

formLayout.addWidget(QLabel("Confirm Password:"))
formLayout.addWidget(self.confirmPassword)

saveBtn = QPushButton("Save")
saveBtn.setFixedHeight(45)
saveBtn.setStyleSheet("""
    QPushButton {
        background-color: #2f8ed2;
        color: white;
        border-radius: 8px;
        font-size: 18px;
    }
    QPushButton:hover {
        background-color: #fdd148;
        color: black;
    }
""")
saveBtn.clicked.connect(self.saveData)

formLayout.addWidget(saveBtn)
formLayout.addStretch()

self.show()

def saveData(self):
    new_username = self.usernameEdit.text().strip()
    new_email = self.gmailEdit.text().strip()
    new_phone = self.phoneEdit.text().strip()

    if new_username:
        self.usernameLabel.setText(f"Username: {new_username}")

    if new_email:
        self.gmailLabel.setText(f"Email: {new_email}")

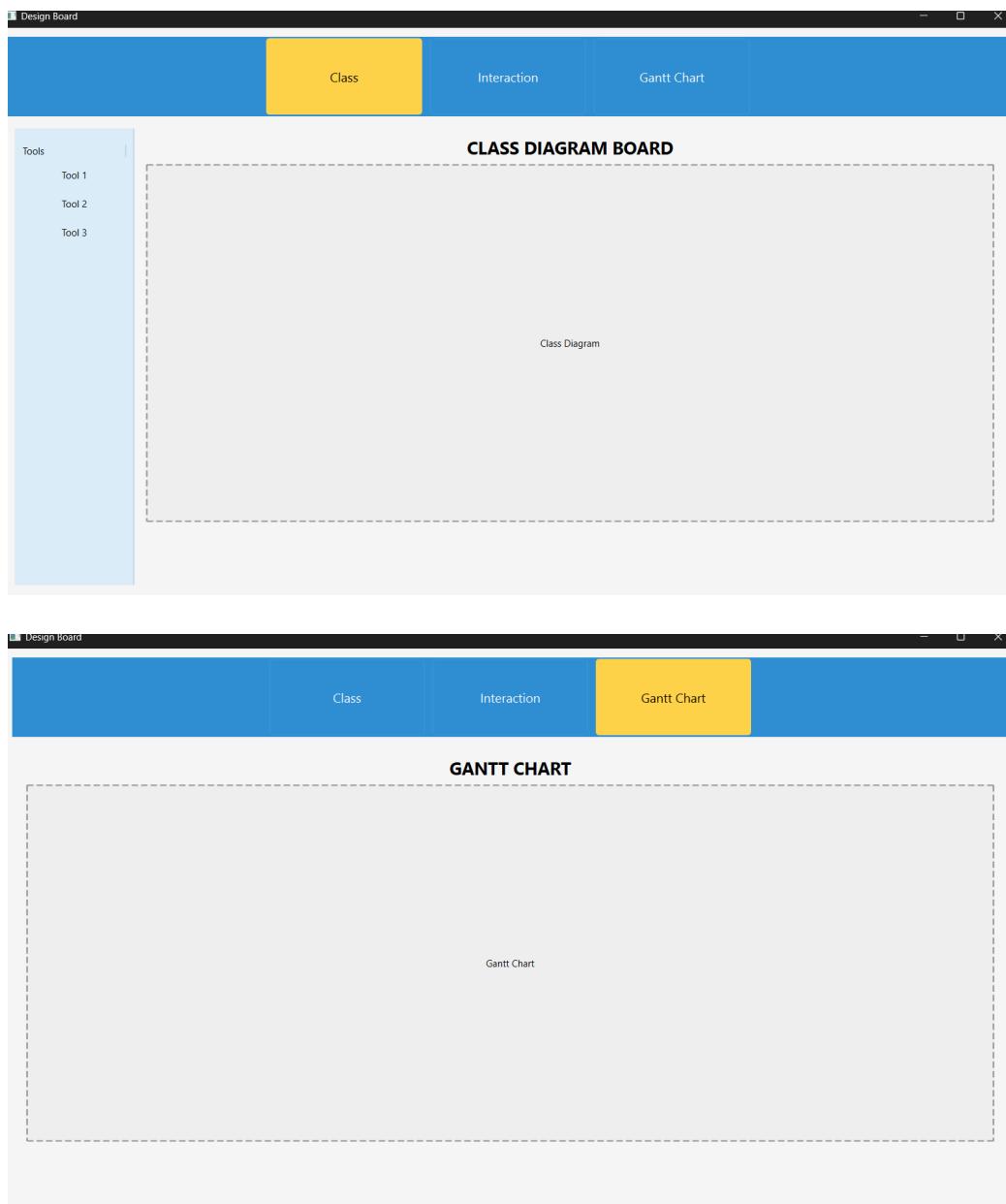
    if new_phone:
        self.phoneLabel.setText(f"Phone: {new_phone}")

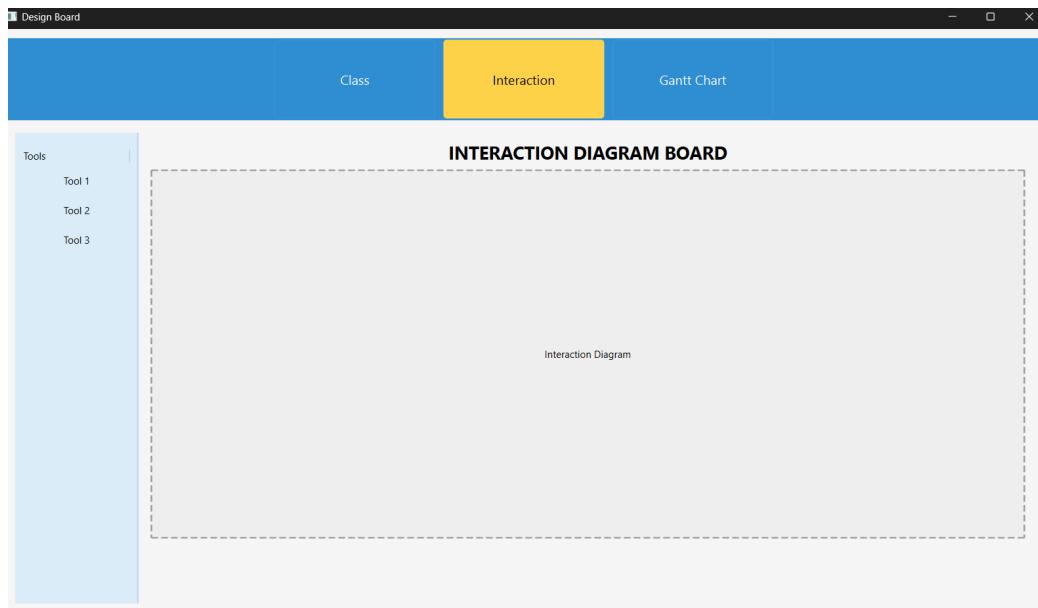
```

```
if __name__ == '__main__':
    app = QApplication(sys.argv)
    w = Setting()
    sys.exit(app.exec())
```

Output:

Main Page



A screenshot of a software application window titled "Setting". The left side features a yellow sidebar with account information:

Account:

Username: Raph
Email: raph@example.com
Phone: +66 1234 5678
Account ID: 12345
Status: Active

The right side contains several input fields and a "Save" button:

Email:

Phone:

Account ID (Read-only):

Change Password

Old Password:

New Password:

Confirm Password:

Save