



Homework 2

Software Engineering Principle
Software Engineering Program,
Department of Computer Engineering,
School of Engineering, KMITL

67011235 Paphavee Yanmook

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, \therefore 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

- **Main Page:** Home page of the program root for navigate to other pages

Code:

MainPage.py

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

lightYellow = "#fef8d7"
yellow = "#fdd148"
blue = "#2f8ed2"
gray = "#3d4355"
white = "#f3f5f4"

class MainPage(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Main Page")
        self.resize(1280, 720)

        # Header
        currProj = "Project 1"
        self.projName = QLabel(currProj, self)
        headFont = QFont("Arial", 24)
        headFont.setBold(True)
        self.projName.setFont(headFont)
        self.projName.setGeometry(355, 25, 337, 35)
        self.projName.setAlignment(Qt.AlignLeft)

        # Username
        self.username = QLabel("Username", self)
        userFont = QFont("Arial", 18)
        self.username.setFont(userFont)
        self.username.setGeometry(1120, 30, 337, 35)
        self.username.setAlignment(Qt.AlignLeft)

        # Project Label
        self.projLabel = QLabel("Projects", self)
        projFont = QFont("Arial", 32)
        projFont.setBold(True)
```

```

self.projLabel.setFont(projFont)
self.projLabel.setGeometry(10, 100, 337, 35)
self.projLabel.setAlignment(Qt.AlignCenter)

# Notification Label
self.notiLabel = QLabel("Notification", self)
subFont = QFont("Arial", 28)
subFont.setBold(True)
self.notiLabel.setFont(subFont)
self.notiLabel.setGeometry(20, 450, 337, 30)
self.notiLabel.setAlignment(Qt.AlignLeft)

# Charter Label
self.projChartLabel = QLabel("Project Charter", self)
subFont.setBold(True)
self.projChartLabel.setFont(subFont)
self.projChartLabel.setGeometry(355, 85, 337, 35)
self.projChartLabel.setAlignment(Qt.AlignLeft)

# Project Objective Label
self.projObjLabel = QLabel("Project Objective", self)
subSubFont = QFont("Arial", 21)
subSubFont.setBold(True)
self.projObjLabel.setFont(subSubFont)
self.projObjLabel.setGeometry(360, 150, 337, 30)
self.projObjLabel.setAlignment(Qt.AlignLeft)

# Project Structure Label
self.projStructLabel = QLabel("Project Structure", self)
self.projStructLabel.setFont(subSubFont)
self.projStructLabel.setGeometry(360, 250, 337, 30)
self.projStructLabel.setAlignment(Qt.AlignLeft)

# Project Member Label
self.projMemLabel = QLabel("Project Member", self)
self.projMemLabel.setFont(subSubFont)
self.projMemLabel.setGeometry(360, 350, 337, 30)
self.projMemLabel.setAlignment(Qt.AlignLeft)

# Project Duration Label
self.projDurLabel = QLabel("Project Duration", self)

```

```

self.projDurLabel.setFont(subSubFont)
self.projDurLabel.setGeometry(360, 450, 337, 30)
self.projDurLabel.setAlignment(Qt.AlignLeft)

# Project Deadline Label
self.projDlLabel = QLabel("Project Deadline", self)
self.projDlLabel.setFont(subSubFont)
self.projDlLabel.setGeometry(360, 550, 337, 30)
self.projDlLabel.setAlignment(Qt.AlignLeft)

# Assignment Label
self.projAssignLabel = QLabel("Assignments", self)
self.projAssignLabel.setFont(subFont)
self.projAssignLabel.setGeometry(755, 85, 337, 35)
self.projAssignLabel.setAlignment(Qt.AlignLeft)

# Projects Buttons
xp = 15
yp = 160
i = 1

for _ in range(5):
    btn = QPushButton("Project " + str(i), self)
    btn.setGeometry(xp, yp, 315, 50)
    btn.setStyleSheet("""
        QPushButton {
            background-color: #f3f5f4;
            border: 1px solid black;
            border-radius: 6px;
            padding: 5px;
        }
        QPushButton:hover {
            background-color: #2f8ed2;
        }
    """)
    yp += 51
    i += 1

# Assignments
xa = 755
ya = 125

```

```

xb = 755
yb = 170
i = 1

for _ in range(7):
    assignmentLabel = QLabel("Task " + str(i), self)
    taskName = QLineEdit(self)
    madeBy = QLabel("By:", self)
    taskOwner = QLineEdit(self)
    statusLabel = QLabel("Status:", self)
    status = QLineEdit(self)
    saveBtn = QPushButton("Save", self)
    saveBtn.setStyleSheet("""
        QPushButton {
            color: "#f3f5f4";
            background-color: #2f8ed2;
            border: 1px solid black;
            border-radius: 6px;
            padding: 5px;
        }
        QPushButton:hover {
            color: "#3d4355";
            background-color: #fdd148;
        }
    """)
    assignmentLabel.setFont(subSubFont)

    assignmentLabel.setGeometry(xa, ya, 315, 50)

    # row elements
    taskName.setGeometry(xb, yb, 200, 30)
    madeBy.setGeometry(xb + 210, yb, 70, 30)
    taskOwner.setGeometry(xb + 235, yb, 100, 30)
    statusLabel.setGeometry(xb + 340, yb, 60, 30)
    status.setGeometry(xb + 390, yb, 50, 30)
    saveBtn.setGeometry(xb + 450, yb, 50, 30)

    ya += 80
    yb += 80
    i += 1

```

```

def paintEvent(self, event):
    painter = QPainter(self)

    painter.setBrush(QColor(lightYellow))

    pen = QPen(QColor(blue), 1)
    painter.setPen(pen)

    mainRect = QRect(5, 5, 1280, 720)
    painter.drawRect(mainRect)

    painter.setBrush(QColor(yellow))
    headerRect = QRect(5, 5, 1280, 66)
    painter.drawRect(headerRect)
    painter.setBrush(QColor(lightYellow))

    projRect = QRect(5, 80, 337, 360)
    painter.drawRect(projRect)

    painter.setBrush(QColor(yellow))
    projTitleRect = QRect(5, 80, 337, 67)
    painter.drawRect(projTitleRect)
    painter.setBrush(QColor(lightYellow))

    notiRect = QRect(5, 436, 337, 289)
    painter.drawRect(notiRect)

    metadataRect = QRect(342, 71, 402, 654)
    painter.drawRect(metadataRect)

    assignRect = QRect(744, 71, 541, 654)
    painter.drawRect(assignRect)

    logoImg = QPixmap("logo-171.png")
    painter.drawPixmap(80, 10, logoImg)

    memberImg = QPixmap("member.png")
    painter.drawPixmap(510, 20, memberImg)

    activeImg = QPixmap("Active.png")
    painter.drawPixmap(1220, 20, activeImg)

```








```

if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = MainPage()
    window.show()
    sys.exit(app.exec())

```

Output:

Main Page

<div>  <div>Project 1</div> <div>    </div> </div> <div>Username </div>		
<div>Projects</div> <div> <div>Project 1</div> <div>Project 2</div> <div>Project 3</div> <div>Project 4</div> <div>Project 5</div> </div>	<div>Project Charter</div> <div>Project Objective</div> <div>Project Structure</div> <div>Project Member</div> <div>Project Duration</div> <div>Project Deadline</div>	<div>Assignments</div> <div>Task 1</div> <div> <input type="text"/> By: <input type="text"/> Status: <input type="text"/> <input type="button" value="Save"/> </div> <div>Task 2</div> <div> <input type="text"/> By: <input type="text"/> Status: <input type="text"/> <input type="button" value="Save"/> </div> <div>Task 3</div> <div> <input type="text"/> By: <input type="text"/> Status: <input type="text"/> <input type="button" value="Save"/> </div> <div>Task 4</div> <div> <input type="text"/> By: <input type="text"/> Status: <input type="text"/> <input type="button" value="Save"/> </div> <div>Task 5</div> <div> <input type="text"/> By: <input type="text"/> Status: <input type="text"/> <input type="button" value="Save"/> </div> <div>Task 6</div> <div> <input type="text"/> By: <input type="text"/> Status: <input type="text"/> <input type="button" value="Save"/> </div> <div>Task 7</div> <div> <input type="text"/> By: <input type="text"/> Status: <input type="text"/> <input type="button" value="Save"/> </div>
Notification		