Homework 9

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:

HW9_67011235_Paphavee.html

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Pyscript</title>
        <link rel="stylesheet" href="https://pyscript.net/releases/2026.1.1/core.css">
        <script type="module" src="https://pyscript.net/releases/2026.1.1/core.js"></script>
    </head>
    <body>
        <section class="pyscript">
            <div id="container"></div>
            <script type="py" src="./HW9_67011235_Paphavee.py"></script>
        </section>
    </body>
</html>
```

HW9_67011235_Paphavee.py

```
import js
from pyscript import document
from pyodide.ffi import create_proxy
from abc import ABC, abstractmethod

white = "white"
lightGray = "#f4f4f4"
yellow = "#ffd77f"
blue = "#32a5da"
darkBlue = "#158bc1"

assets = {
    "logoImg": "images/logo.png",
    "membersImg": "images/member.png",
    "userImg": "images/active.png",
    "saveSound": "sounds/save.wav",
}

class AbstractWidget(ABC):
    def __init__(self, element_id):
        self.element_id = element_id
        self._element = None

    @property
    def element(self):
        if not self._element:
            self._element =
              ↪  document.querySelector(f"#{self.element_id}")
        return self._element

    @abstractmethod
    def drawWidget(self):
        pass

class MainPage(AbstractWidget):
    def __init__(self, element_id):
        super().__init__(element_id)
        self._proxies = []
```

```
def clearElement(self, el):
    while el.firstChild:
        el.removeChild(el.firstChild)

def saveProjCharter(self, evt=None):
    self.saveBtnSound =
       ↪  js.Audio.new(assets["saveSound"]).play()
    data = {
        "objective": self.projObjInput.value,
        "structure": self.projStructInput.value,
        "members": self.projMembersInput.value,
        "duration": self.projDurationInput.value,
        "deadline": self.projDeadlineInput.value,
    }

def saveAssignment(self, evt=None):
    self.saveBtnSound =
       ↪  js.Audio.new(assets["saveSound"]).play()
    data = {
        "task": self.taskInput.value,
        "person": self.personInput.value,
        "status": self.statusInput.value
    }

def makeCharterRow(self, parent, label_text):
    group = document.createElement("div")
    group.style.display = "flex"
    group.style.flexDirection = "column"
    group.style.padding = "10px"
    group.style.gap = "6px"
    parent.appendChild(group)

    title = document.createElement("h2")
    title.textContent = label_text
    title.style.margin = "0"

    row = document.createElement("div")
    row.style.display = "flex"
    row.style.gap = "10px"
    row.style.alignItems = "center"
```

```
        inp = document.createElement("input")
        inp.type = "text"
        inp.style.flex = "1"
        inp.style.minWidth = "0"
        inp.style.maxWidth = "390px"
        inp.style.padding = "10px"
        inp.style.fontSize = "16px"
        inp.style.border = "2px solid " + blue
        inp.style.borderRadius = "6px"
        inp.style.boxSizing = "border-box"

        row.appendChild(inp)
        group.appendChild(title)
        group.appendChild(row)

        return inp

    def drawProjectCharter(self):
        self.projectDetails.style.display = "flex"
        self.projectDetails.style.flexDirection = "row"
        self.projectDetails.style.gap = "10px"
        self.projectDetails.style.justifyContent = "flex-start"
        self.projectDetails.style.alignItems = "stretch"

        self.clearElement(self.projectDetails)

        metadata = document.createElement("div")
        metadata.style.flex = "0 0 390px"
        metadata.style.display = "flex"
        metadata.style.flexDirection = "column"
        metadata.style.border = "2px solid " + blue
        metadata.style.borderRadius = "12px"
        metadata.style.boxSizing = "border-box"
        metadata.style.padding = "10px"
        metadata.style.margin = "10px"
        metadata.style.gap = "12px"
        self.projectDetails.appendChild(metadata)

        projectCharterTitle = document.createElement("h1")
        projectCharterTitle.textContent = "Project Charter"
        projectCharterTitle.style.height = "70px"
```

```
projectCharterTitle.style.display = "flex"
projectCharterTitle.style.justifyContent = "center"
projectCharterTitle.style.alignItems = "center"
projectCharterTitle.style.margin = "0"
projectCharterTitle.style.backgroundColor = yellow
projectCharterTitle.style.border = "2px solid " + blue
projectCharterTitle.style.borderRadius = "12px"
projectCharterTitle.style.boxSizing = "border-box"
metadata.appendChild(projectCharterTitle)

projectCharterContent = document.createElement("div")
projectCharterContent.style.display = "flex"
projectCharterContent.style.flexDirection = "column"
projectCharterContent.style.gap = "28px"
projectCharterContent.style.flex = "1"
projectCharterContent.style.overflowY = "auto"
metadata.appendChild(projectCharterContent)

self.projObjInput =
   ↪  self.makeCharterRow(projectCharterContent, "Project
   ↪  Objective")
self.projStructInput =
   ↪  self.makeCharterRow(projectCharterContent, "Project
   ↪  Structure")
self.projMembersInput =
   ↪  self.makeCharterRow(projectCharterContent, "Project
   ↪  Member(s)")
self.projDurationInput =
   ↪  self.makeCharterRow(projectCharterContent, "Project
   ↪  Duration")
self.projDeadlineInput =
   ↪  self.makeCharterRow(projectCharterContent, "Project
   ↪  Deadline")

saveProjCharterBtn = document.createElement("button")
saveProjCharterBtn.textContent = "Save All"
saveProjCharterBtn.className = "btn"
proxy = create_proxy(self.saveProjCharter)
saveProjCharterBtn.addEventListener("click", proxy)
self._proxies.append(proxy)
projectCharterContent.appendChild(saveProjCharterBtn)
```

```
def makeAssignmentRow(self, parent, index):
    card = document.createElement("div")
    card.style.padding = "10px"

    row = document.createElement("div")
    row.style.display = "flex"
    row.style.alignItems = "center"
    row.style.gap = "10px"
    row.style.width = "100%"
    row.style.flexWrap = "nowrap"

    taskLabel = document.createElement("span")
    taskLabel.textContent = "Task " + str(index + 1)
    taskLabel.style.width = "70px"
    taskLabel.style.flex = "0 0 auto"

    self.taskInput = document.createElement("input")
    self.taskInput.type = "text"
    self.taskInput.placeholder = "Task name"
    self.taskInput.style.flex = "2"
    self.taskInput.style.minWidth = "0"
    self.taskInput.style.padding = "8px"
    self.taskInput.style.border = "2px solid " + blue
    self.taskInput.style.borderRadius = "6px"
    self.taskInput.style.boxSizing = "border-box"

    byLabel = document.createElement("span")
    byLabel.textContent = "By:"
    byLabel.style.flex = "0 0 auto"

    self.personInput = document.createElement("input")
    self.personInput.type = "text"
    self.personInput.placeholder = "Member"
    self.personInput.style.flex = "1"
    self.personInput.style.minWidth = "0"
    self.personInput.style.padding = "8px"
    self.personInput.style.border = "2px solid " + blue
    self.personInput.style.borderRadius = "6px"
    self.personInput.style.boxSizing = "border-box"
```

```
        statusLabel = document.createElement("span")
        statusLabel.textContent = "Status:"
        statusLabel.style.flex = "0 0 auto"

        self.statusInput = document.createElement("select")
        self.statusInput.style.flex = "0 0 150px"
        self.statusInput.style.padding = "8px"
        self.statusInput.style.border = "2px solid " + blue
        self.statusInput.style.borderRadius = "6px"
        self.statusInput.style.boxSizing = "border-box"

        for s in ["Not Started", "In Progress", "Completed"]:
            opt = document.createElement("option")
            opt.textContent = s
            self.statusInput.appendChild(opt)

        saveAssignmentBtn = document.createElement("button")
        saveAssignmentBtn.textContent = "Save"
        saveAssignmentBtn.className = "btn"
        proxy = create_proxy(self.saveAssignment)
        saveAssignmentBtn.addEventListener("click", proxy)
        self._proxies.append(proxy)

        row.appendChild(taskLabel)
        row.appendChild(self.taskInput)
        row.appendChild(byLabel)
        row.appendChild(self.personInput)
        row.appendChild(statusLabel)
        row.appendChild(self.statusInput)
        row.appendChild(saveAssignmentBtn)
        card.appendChild(row)
        parent.appendChild(card)


    def drawProjectAssignments(self):
        assignments = document.createElement("div")
        assignments.style.flex = "1"
        assignments.style.display = "flex"
        assignments.style.flexDirection = "column"
        assignments.style.gap = "12px"
        assignments.style.border = "2px solid " + blue
        assignments.style.borderRadius = "12px"
        assignments.style.boxSizing = "border-box"
        assignments.style.padding = "10px"
        assignments.style.margin = "10px"
        self.projectDetails.appendChild(assignments)

        assignmentsTitle = document.createElement("h1")
        assignmentsTitle.textContent = "Assignments"
        assignmentsTitle.style.height = "70px"
        assignmentsTitle.style.display = "flex"
        assignmentsTitle.style.justifyContent = "center"
        assignmentsTitle.style.alignItems = "center"
        assignmentsTitle.style.margin = "0"
        assignmentsTitle.style.backgroundColor = yellow
        assignmentsTitle.style.border = "2px solid " + blue
        assignmentsTitle.style.borderRadius = "12px"
        assignments.appendChild(assignmentsTitle)

        assignmentsContent = document.createElement("div")
        assignmentsContent.style.display = "flex"
        assignmentsContent.style.flex = "1"
        assignmentsContent.style.overflowY = "auto"
        assignmentsContent.style.flexDirection = "column"
        assignmentsContent.style.gap = "20px"
        assignments.appendChild(assignmentsContent)

        for i in range(10):
            self.makeAssignmentRow(assignmentsContent, i)

    def onClickProjectBtn(self, i):
        self.currProj.textContent = "Current Project: Project " +
           ↪  str(i + 1)
        self.drawProjectCharter()
        self.drawProjectAssignments()

    def drawWidget(self):
        style = document.createElement("style")
        style.textContent = f"""
        .btn {{
            padding: 10px 14px;
            font-size: 16px;
            border: 2px solid {blue};
            border-radius: 6px;
            background-color: {white};
            cursor: pointer;
            transition: 0.15s;
        }}

        .btn:hover {{
            background-color: {yellow};
            transform: translateY(-2px);
            box-shadow: 0 4px 8px rgba(0,0,0,0.15);
        }}

        .btn:active {{
            transform: translateY(0px);
            box-shadow: none;
        }}
        """
        document.head.appendChild(style)

        document.body.style.backgroundColor = lightGray
        document.body.style.margin = "0"
        document.body.style.fontFamily = "Arial, sans-serif"

        while self.element.firstChild:
            self.element.removeChild(self.element.firstChild)

        container = document.createElement("div")
        container.style.display = "flex"
        container.style.flexDirection = "column"
        container.style.height = "100vh"
        container.style.gap = "10px"
        container.style.padding = "10px"
        container.style.boxSizing = "border-box"
        self.element.appendChild(container)

        headerRow = document.createElement("div")
        headerRow.style.display = "flex"
        headerRow.style.width = "100%"
        headerRow.style.height = "80px"
        headerRow.style.boxSizing = "border-box"
```

```
container.appendChild(headerRow)

logoBox = document.createElement("div")
logoBox.style.width = "400px"
logoBox.style.height = "80px"
logoBox.style.backgroundColor = blue
logoBox.style.display = "flex"
logoBox.style.justifyContent = "center"
logoBox.style.alignItems = "center"
logoBox.style.boxSizing = "border-box"
logoBox.style.border = "2px solid " + blue
logoBox.style.borderRadius = "12px 0 0 12px"

logo = document.createElement("img")
logo.src = "./" + assets["logoImg"]
logo.style.height = "70px"
logo.style.objectFit = "contain"
logoBox.appendChild(logo)

navBar = document.createElement("div")
navBar.style.flex = "1"
navBar.style.height = "80px"
navBar.style.display = "flex"
navBar.style.alignItems = "center"
navBar.style.padding = "0 20px"
navBar.style.backgroundColor = blue
navBar.style.border = "2px solid " + blue
navBar.style.borderLeft = "0"
navBar.style.borderRadius = "0 12px 12px 0"
navBar.style.boxSizing = "border-box"
navBar.style.color = white

self.currProj = document.createElement("h1")
self.currProj.textContent = "Current Project:"
self.currProj.style.margin = "0"
self.currProj.style.fontSize = "20px"
self.currProj.style.fontWeight = "700"

spacer = document.createElement("div")
spacer.style.flex = "1"
```

```
members = document.createElement("img")
members.src = "./" + assets["membersImg"]
members.style.height = "36px"
members.style.objectFit = "contain"

usernameText = document.createElement("h2")
usernameText.textContent = "Username"
usernameText.style.margin = "0 0 0 12px"
usernameText.style.fontSize = "20px"
usernameText.style.fontWeight = "400"

userProfile = document.createElement("img")
userProfile.src = "./" + assets["userImg"]
userProfile.style.height = "36px"
userProfile.style.width = "36px"
userProfile.style.objectFit = "contain"
userProfile.style.marginLeft = "6px"

navBar.appendChild(self.currProj)
navBar.appendChild(spacer)
navBar.appendChild(members)
navBar.appendChild(usernameText)
navBar.appendChild(userProfile)

headerRow.appendChild(logoBox)
headerRow.appendChild(navBar)

contentRow = document.createElement("div")
contentRow.style.display = "flex"
contentRow.style.flex = "1"
contentRow.style.gap = "10px"
contentRow.style.boxSizing = "border-box"
container.appendChild(contentRow)

leftSideBar = document.createElement("div")
leftSideBar.style.display = "flex"
leftSideBar.style.flexDirection = "column"
leftSideBar.style.width = "400px"
leftSideBar.style.gap = "10px"
leftSideBar.style.boxSizing = "border-box"
```

```
projects = document.createElement("div")
projects.style.display = "flex"
projects.style.flex = "1"
projects.style.flexDirection = "column"
projects.style.border = "2px solid " + blue
projects.style.borderRadius = "12px"
projects.style.boxSizing = "border-box"
projects.style.gap = "20px"
projects.style.padding = "10px"

projectsTitle = document.createElement("h1")
projectsTitle.textContent = "Projects"
projectsTitle.style.height = "70px"
projectsTitle.style.display = "flex"
projectsTitle.style.justifyContent = "center"
projectsTitle.style.alignItems = "center"
projectsTitle.style.margin = "0"
projectsTitle.style.backgroundColor = yellow
projectsTitle.style.border = "2px solid " + blue
projectsTitle.style.borderRadius = "12px"
projectsTitle.style.boxSizing = "border-box"

projectsList = document.createElement("div")
projectsList.style.display = "flex"
projectsList.style.flexDirection = "column"
projectsList.style.gap = "20px"

for i in range(6):
    projectButton = document.createElement("button")
    projectButton.textContent = "Project " + str(i + 1)
    projectButton.className = "btn"

    proxy = create_proxy(lambda evt, x=i:
       ↪  self.onClickProjectBtn(x))
    projectButton.addEventListener("click", proxy)
    self._proxies.append(proxy)

    projectsList.appendChild(projectButton)

projects.appendChild(projectsTitle)
projects.appendChild(projectsList)
```

```
noti = document.createElement("div")
noti.style.flex = "1"
noti.style.display = "flex"
noti.style.flexDirection = "column"
noti.style.border = "2px solid " + blue
noti.style.borderRadius = "12px"
noti.style.boxSizing = "border-box"
noti.style.padding = "10px"
noti.style.gap = "10px"

notiTitle = document.createElement("h1")
notiTitle.textContent = "Notifications"
notiTitle.style.height = "70px"
notiTitle.style.display = "flex"
notiTitle.style.justifyContent = "center"
notiTitle.style.alignItems = "center"
notiTitle.style.margin = "0"
notiTitle.style.backgroundColor = yellow
notiTitle.style.border = "2px solid " + blue
notiTitle.style.borderRadius = "12px"
notiTitle.style.boxSizing = "border-box"

notiBody = document.createElement("textarea")
notiBody.style.flex = "1"
notiBody.style.resize = "none"
notiBody.style.borderRadius = "10px"
notiBody.style.padding = "10px"
notiBody.style.boxSizing = "border-box"

noti.appendChild(notiTitle)
noti.appendChild(notiBody)

leftSideBar.appendChild(projects)
leftSideBar.appendChild(noti)

rightSideBar = document.createElement("div")
rightSideBar.style.display = "flex"
rightSideBar.style.flex = "1"
rightSideBar.style.flexDirection = "column"
rightSideBar.style.gap = "10px"
```

```
        rightSideBar.style.boxSizing = "border-box"

        self.projectDetails = document.createElement("div")
        self.projectDetails.style.flex = "1"
        self.projectDetails.style.border = "2px solid " + blue
        self.projectDetails.style.borderRadius = "12px"
        self.projectDetails.style.boxSizing = "border-box"
        self.projectDetails.style.display = "flex"
        self.projectDetails.style.justifyContent = "center"
        self.projectDetails.style.alignItems = "center"

        self.projectDetailsText = document.createElement("h1")
        self.projectDetailsText.textContent = "Please select a project first"
        self.projectDetailsText.style.margin = "0"
        self.projectDetails.appendChild(self.projectDetailsText)

        rightSideBar.appendChild(self.projectDetails)

        contentRow.appendChild(leftSideBar)
        contentRow.appendChild(rightSideBar)


if __name__ == "__main__":
    output = MainPage("container")
    output.drawWidget()
```

Image:

`logo.png`

`active.png`

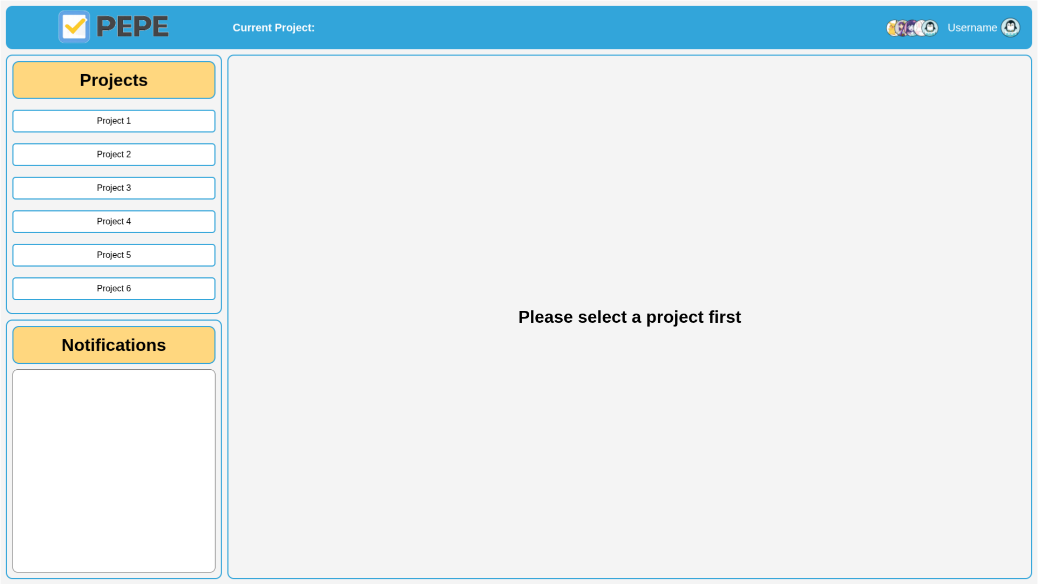
`member.png`

Sound:

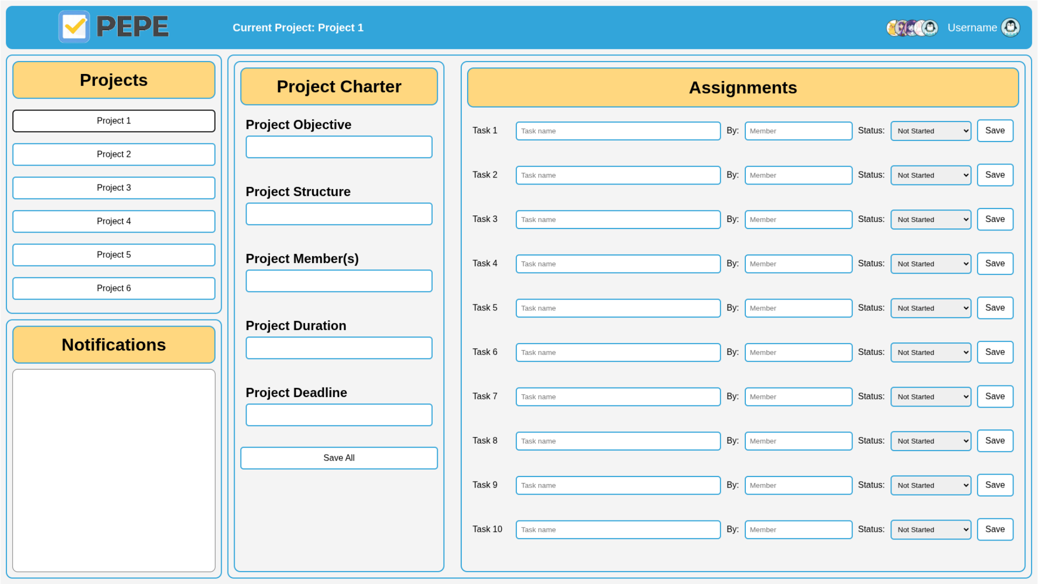
- `save.wav`

Output:

Main Page



Please select a project first



Task 1	Task name	By:	Member	Status:	Not Started	Save
Task 2	Task name	By:	Member	Status:	Not Started	Save
Task 3	Task name	By:	Member	Status:	Not Started	Save
Task 4	Task name	By:	Member	Status:	Not Started	Save
Task 5	Task name	By:	Member	Status:	Not Started	Save
Task 6	Task name	By:	Member	Status:	Not Started	Save
Task 7	Task name	By:	Member	Status:	Not Started	Save
Task 8	Task name	By:	Member	Status:	Not Started	Save
Task 9	Task name	By:	Member	Status:	Not Started	Save
Task 10	Task name	By:	Member	Status:	Not Started	Save