

```

"""
Ryan Kennedy, Gabriel Waldner
Cmdr. Schenk
Cloud Computing
7th Period
May 5, 2025
"""

import sys
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox

import bcrypt

from user_record import UserRecord
from task_record import TaskRecord
from database import Database
from gui_states import GUIStates

class UserInfo:

    def __init__(self, root, db):
        self.frame = tk.Frame(root)
        self.line_canvas = tk.Canvas(self.frame, width=800, height=800, bg="white", highlightthickness=0)
        self.db = db
        self.user = UserRecord() # stores the user that we are currently displaying the information of, currently blank but gets set in self.show(...)

    def init_resources(self):
        style = ttk.Style()
        style.theme_use('clam')

        # Title Label
        style.configure("TitleLabel.TLabel", font=("Segoe UI", 20, "italic"))

        style.configure("HeaderLabel.TLabel", font=("Segoe UI", 12, "italic"))

        # Field Labels
        style.configure("FieldLabel.TLabel", font=("Segoe UI", 10), foreground="#444", background="white")

        # Field Entry
        style.configure("Field.TEntry", font=("Segoe UI", 7), padding=10)

        # Info Buttons
        style.configure("Info.TButton", font=("Segoe UI", 7), padding=10)

        # Action Buttons
        style.configure("Action.TButton", font=("Segoe UI", 10), padding=10)

        # Delete Buttons
        style.configure("Delete.TButton", font=("Segoe UI", 20), padding=10, background="#d32f2f")

        ttk.Label(self.frame, text="Account Info", style="TitleLabel.TLabel").place(x=400, y=30, anchor="center")

        # change name
        ttk.Label(self.frame, text="Change Name", style="HeaderLabel.TLabel").place(x=20, y=60)

        ttk.Label(self.frame, text="Name:", style="FieldLabel.TLabel").place(x=100, y=125)
        self.name_ent = ttk.Entry(self.frame, style="Field.TEntry")

```

```

        self.name_ent.place(x=100, y=150)

        ttk.Button(self.frame, text="Save", command=self.change_name, style="Info.TButton").place(x=100, y=225, width=150, height=50)

        # change credentials

        ttk.Label(self.frame, text="Change Credentials", style="HeaderLabel.TLabel").place(x=20, y=350)

        ttk.Label(self.frame, text="Username:", style="FieldLabel.TLabel").place(x=100, y=400)
        self.username_ent = ttk.Entry(self.frame, style="Field.TEntry")
        self.username_ent.place(x=100, y=425)

        ttk.Label(self.frame, text="New Password:", style="FieldLabel.TLabel").place(x=100, y=500)
        self.password_ent = ttk.Entry(self.frame, style="Field.TEntry")
        self.password_ent.place(x=100, y=525)

        ttk.Label(self.frame, text="Old Password:", style="FieldLabel.TLabel").place(x=100, y=600)
        self.old_password_ent = ttk.Entry(self.frame, style="Field.TEntry")
        self.old_password_ent.place(x=100, y=625)

        ttk.Button(self.frame, text="Save", command=self.change_credentials, style="Info.TButton").place(x=100, y=700, width=150, height=50)

        # buttons for action

        ttk.Button(self.frame, text="Goto Task Browser", command=self.goto_task_browser, style="Action.TButton").place(x=500, y=80, width=225, height=100)
        ttk.Button(self.frame, text="Logout", command=self.logout, style="Action.TButton").place(x=500, y=280, width=225, height=100)
        ttk.Button(self.frame, text="Quit App", style="Action.TButton", command=lambda: sys.exit(0)).place(x=500, y=466, width=225, height=100)
        ttk.Button(self.frame, text="Delete Account", command=self.delete_user, style="Delete.TButton").place(x=500 - 8, y=650, width=240, height=100)

    def show(self, id):
        self.clear_entries()
        self.user = self.db.users_get_record_by_id(id)
        self.name_ent.insert(0, self.user.name)
        self.username_ent.insert(0, self.user.username)
        self.line_canvas.place(x=0, y=0)
        self.frame.place(x=0, y=0, width=800, height=800)

        self.line_canvas.create_line(400, 75, 400, 800, fill="black", width=5)

    def delete_user(self):
        result = messagebox.askquestion("Account Deletion Confirmation", "Are you sure that you would like to delete your account?")
        if (result != "yes"):
            return

        self.db.users_delete(self.user.id)
        self.goto_login()

    def change_credentials(self):
        result = messagebox.askquestion("Change Credentials Confirmation", "Are you sure that you would like to update your credentials?")
        if (result != "yes"):
            return

        old_password = self.old_password_ent.get()
        hashed_old_password = bcrypt.hashpw(old_password.encode(), b'$2b$12$zm4/D56Nt1i/hWPKnmLSgu').decode("utf-8")

```

```

        if (hashed_old_password != self.user.hashed_password):
            tk.messagebox.showerror("Failed to change credentials", "Old Password is I
ncorrect")
            return

        password = self.password_ent.get()
        hashed_password = bcrypt.hashpw(password.encode(), b'$2b$12$zm4/D56Ntli/hWPKnm
LSgu').decode("utf-8")

        record = UserRecord()
        record.fill(id=self.user.id, name=self.name_ent.get(), username=self.username_
ent.get(), hashed_password=hashed_password)
        self.db.users_update(record)
        self.user = self.db.users_get_record_by_id(self.user.id)
        self.update_entries()
        tk.messagebox.showinfo("Successfully Updated Credentials", "Successfully Updat
ed Credentials")

    def change_name(self):
        record = UserRecord()
        record.fill(id=self.user.id, name=self.name_ent.get(), username=self.user.user
name, hashed_password=self.user.hashed_password)
        self.db.users_update(record)
        tk.messagebox.showinfo("Successfully Updated Name", "Successfully Updated Name
")

        self.user = self.db.users_get_record_by_id(self.user.id)
        self.update_entries()

    def update_entries(self):
        self.clear_entries()
        self.name_ent.insert(0, self.user.name)
        self.username_ent.insert(0, self.user.username)

    def clear_entries(self):
        self.name_ent.delete(0, "end")
        self.username_ent.delete(0, "end")
        self.password_ent.delete(0, "end")
        self.old_password_ent.delete(0, "end")

    def logout(self):
        self.goto_login()

    def goto_login(self):
        self.hide()
        self.show_map[GUIStates.LOGIN_USER]()

    def goto_task_browser(self):
        self.hide()
        self.show_map[GUIStates.TASK_BROWSER](self.user.id)

    def hide(self):
        self.line_canvas.place_forget()
        self.frame.place_forget()

    def assign_show_map(self, show_map):
        self.show_map = show_map

```